Egylépéses Numerikus Módszerek Közönséges Differenciálegyenletek Megoldására

Diplomamunka

írta: Turi Petra

Matematika BSc szak

Témavezető:

Szekeres Béla János, tudományos segédmunkatárs
Alkalmazott Analízis és Számításmatematikai Tanszék
Eötvös Loránd Tudományegyetem, Természettudományi Kar

Eötvös Loránd Tudományegyetem
Természettudományi Kar
2016

Köszönetnyilvánítás

Elsősorban szeretném megköszönni témavezetőmenk Szekeres Béla Jánosnak odaadó, kivételesen lelkiismeretes munkáját és türelmét, amellyel végig kísérte dolgozatom elkészítését.

A dolgozatom alapjait jelentő MATLAB kódok megírásában és ellenőrzésében nagyban segítségemre volt a témavezetőmön túl Csajbók Viktória, a Budapesti Műszaki egyetem doktorandusza. Ezúton szeretném neki megköszönni segítségét és türelmét, amellyel hozzájárult, hogy ez a dolgozat elkészülhessen.

Végül, de nem utolsó sorban köszönet illeti családomat feltétel nélküli támogatásukért és segítségükért, valamint barátaimat bíztatásukért és rendkívül pozitív hozzáállásukért.

Budapest, 2016 május 31

Turi Petra

Tartalomjegyzék

1. fejezet

Bevezetés

Dolgozatom témájául közönséges differenciálegyenletek numerikus megoldását választottam. Az általunk vizsgált Cauchy-feladat a következő alakú:

$$\frac{du}{dt} = f(t,u), \quad t \in [0,T]$$
$$u(0) = u_0,$$

ahol u_0 adott kezdeti függvény. A differenciálegyenlet megoldása azt jelenti, hogy megkeressük azt az u függvényt, amely a fenti egyenletet igazzá teszi. A gyakorlatban előforduló problémák esetén ezt a függvényt ritkán tudjuk zárt alakban kifejezni, ezért numerikusan közelítjük.

A dolgozatban az [1]-es munka 3. és 4. fejezetének néhány definíciójának, tételének, illetve algoritmusának áttekintése szerepel. Az első fejezetben ismertetjük a legalapvetőbb egyszerű egylépéses numerikus módszereket: a θ-módszert és a Runge-Kutta módszercsaládot. A második fejezetben pedig különböző feladatokon vizsgáljuk a módszerek működését, mint a redukált háromtest-probléma, egy egyszerű ragadozó-zsákmány modell, valamint a klasszikus Lorenz-féle differenciálegyenlet-rendszer, ami a differenciálegyenletek elméletének fejlődésében kulcsszerepet töltött be.

A numerikus módszerek implementálásához MATLAB programkörnyezetet használtunk, a módszereket megvalósító kódokat szintén ismertetjük.

2. fejezet

A szükséges előismeretek összefoglalása

2.1. Közönséges differenciálegyenletek

A differenciálegyenleteket többségében a természettudományok, mérnöki, és közgazdasági folyamatok leírására használják, azaz a folytonos matematikai modelleket többnyire ezek segítségével lehetséges (és szokásos) leírni. Több fajtáját különböztetjük meg, például a közönséges (egyváltozós az ismeretlen függvény), és a parciális differenciálegyenleteket.
Néhány fontosabb jellemzőjük:

- **rend**: egy differenciálegyenlet rendje a differenciálegyenletben szereplő legmagasabb rendű derivált rendje,

- **explicit**: egy differenciálegyenlet explicit, ha a függvénykapcsolatból explicit kifejezhető a legmagasabb rendű derivált,

- **implicit**: egy differenciálegyenlet implicit, ha nem explicit, tehát nem tudjuk kifejezni a legmagasabb rendű deriváltat.

2.1.1. Definíció. *Legyen $G \subset \mathbb{R} \times \mathbb{R}^d$ egy tartomány (azaz összefüggő, nyílt halmaz), $(t_0, \bar{u_0}) \subset$ $\subset G$ egy adott pont ($t_0 \subset \mathbb{R}$, $\bar{u_0} \subset \mathbb{R}^d$), $f : G \to \mathbb{R}^d$ egy folytonos leképezés. A*

$$\begin{cases} \frac{du(t)}{dt} = f(t,u) \\ u(t_0) = \bar{u_0} \end{cases}$$

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

Amikor a Cauchy-feladat egy természettudományos, műszaki vagy közgazdasági folyamat matematikai modellje, akkor alapvető követelmény, hogy létezzen egyértelmű megoldása. Ezt biztosítja az alábbi tétel a [2] munkából:

2.1.2. Tétel. *(egzisztenciatétel): Tegyük fel, hogy teljesülnek az alábbi feltételek:*

- $f : T \to \mathbb{R}$, $T \subset \mathbb{R}^2$ *egy tartomány,*
- $f \subset C(T)$, *azaz* f *folytonos* T*-n,*
- f *a második változójában eleget tesz a Lipschitz-feltételnek, azaz létezik olyan* $L > 0$ *állandó, hogy minden* (t, u_1), $(t, u_2) \subset T$*-re igaz, hogy*

$$|f(t, u_1) - f(t, u_2)| \leq L|u_1 - u_2|.$$

Ekkor a kezdeti feltételbeli t_0 *pontnak létezik olyan* $K(t_0)$ *környezete, hogy a kezdetiérték feladatnak egyértelműen létezik megoldása* $K(t_0)$*-n.*

2.2. Numerikus módszerek

Sajnos csak ritkán lehetséges a differenciálegyenletek megoldásának konkrét előállítása zárt alakban (azaz megadása olyan képletek segítségével, amelyek ismert és könnyen kiértékelhető függvényeket tartalmaznak), ezért gyakorlatilag az a legegyszerűbb, ha numerikus módszerek segítségével valamilyen közelítő alakban keressük őket. Természetesen ezek a módszerek csak közelítő megoldást adnak, de kellően nagy pontosságúak, és ellenőrizhető hibahatáron belül mozognak. Ennek érdekében különböző elvárásokkal élünk velük szemben:

* Létezzen megoldás (egzisztencia).
* Ez a megoldás egyértelmű legyen (unicitás).
* A megoldás folytonosan függjön a feladatot leíró adatoktól (stabilitás).

A feladatokat, amelyek tudják a felsorolt elvárásokat, *korrekt kitűzésűeknek* nevezzük.
Sajnos többnyire nagyon speciális f függvények esetén adhatók csak meg képletek segítségével a kezdetiérték-feladatok megoldásai. Ezért ezek helyett inkább *numerikus megoldást* állítunk elő, ami azt jelenti, hogy az értelmezési tartományának egyes pontjaiban az ismeretlen megoldásfüggvény értékeit véges számú lépéssel közelítőleg határozzuk meg.
Ezután ebben a dolgozatban az úgynevezett *egylépéses módszerek*kel és az általuk megkapható közelítő megoldásokkal fogunk foglalkozni, vagyis olyan típusú eljárásokkal, ahol valamely rögzített időpontbeli közelítést egy az azt megelőző időpontbeli közelítés felhasználásával határozunk

meg. A továbbiakban az alábbi feladatot vizsgáljuk:

$$\begin{aligned} &\frac{du}{dt} = f(t,u); \quad t \in [0,T] \\ &u(0) = u_0 \end{aligned} \tag{2.1}$$

ahol $T > 0$ olyan szám, amely mellett a feladatnak létezik egyértelmű, megfelelően sima megoldása a $[0,T]$ intervallumon.

Szakdolgozatom témája az előbbi (2.1) egyenlet numerikus megoldása különböző tesztfeladatokon. Ehhez egylépéses módszereket fogunk használni. A következőkben ezeket a módszereket definiáljuk és vizsgáljuk az [1]-es munka alapján. Az egylépéses módszereket az úgynevezett ekvidisztáns rácshálón (illetve azok sorozatán) adjuk meg. Ilyen rácsháló:

$$\omega_h := \left\{ t_i = ih; \quad i = 0,1,\cdots,N; \quad h = \frac{T}{N} \right\}.$$

A numerikus módszereket y_i függvényekkel adjuk meg, mellyel célunk, hogy $y_i \approx u(t_i)$, ahol $u(t_i)$ a pontos megoldás az ω_h rács i. pontjában. Ezek után bevezethetjük a numerikus módszer definícióját:

2.2.1. Definíció. *Legyen*

$$y_{i+1} = y_i + h\Phi(h, t_i, y_i, y_{i+1}), \tag{2.2}$$

ahol Φ adott függvény és $i \in \mathbb{N}$, továbbá $t_i \in \omega_h$ és $y_0 = u_0$. A továbbiakban azt a numerikus módszert, amelyet a fenti képlet realizál, Φ numerikus módszernek nevezzük.

2.2.1. A θ-módszer

Első lépésként olyan $P_1(t)$ elsőfokú polinomot keresünk, amely mellett az

$$|u(t_{i+1}) - P_1(t_{i+1})| = O(h_i^2) \tag{2.3}$$

becslés érvényes. Ezért $P_1(t)$-től elvárjuk, hogy menjen keresztül a $(t_i, u(t_i))$ ponton, és irányát az $u(t)$ függvény t_i és t_{i+1} pontbeli érintőinek iránya határozza meg. (Ugyanis az irányt a megoldásgörbe rácspontbeli értékeiből akartuk meghatározni.)
Ezért legyen

$$P_1(t) := u(t_i) + \alpha(t - t_i) \quad (t \in [t_i, t_{i+1}]) \tag{2.4}$$

alakú, ahol $\alpha = \alpha(u_0(t_i), u_0(t_{i+1}))$ egy adott függvény. (Például az $\alpha = u_0(t_i)$ megválasztással $P_1(t) = T_{1,u}(t)$, és ekkor természetesen érvényes.)
Létezik-e más, megfelelő megválasztás is?
Mivel

$$u(t_{i+1}) = u(t_i) + u_0(t_i)h_i + O(h_i^2) \tag{2.5}$$

ebből következik, hogy

$$u(t_{i+1}) - P_1(t_{i+1}) = h_i(u_0(t_i) - \alpha) + O(h_i^2), \tag{2.6}$$

tehát pontosan akkor teljesül, amikor az

$$\alpha - u_0(t_i) = O(h_i) \tag{2.7}$$

becslés fennáll.

2.2.2. Tétel. *Ekkor tetszőleges $\theta \in R$ esetén az $\alpha = (1-\theta)u'_{t_i} + \theta u'_{t_{i+1}}$ megválasztású α függvény esetén az előbbi becslés érvényes.*

A fenti $P_1(t)$ polinom az

$$y_{i+1} = y_i + \alpha h_i \tag{2.8}$$

egylépéses numerikus módszert határozza meg, ahol (2.1) összefüggés és (2.2.2) tétel alapján

$$\alpha = (1-\theta)f(t_i, y_i) + \theta f(t_{i+1}, y_{i+1}). \tag{2.9}$$

Az itt leírt módszert θ-módszernek nevezzük.

2.2.3. Megjegyzés. *A θ-módszer esetén is jellemző, hogy y_i valamilyen közelítése az $u(t_i)$ pontos értéknek. Az eltérés alapvetően a következő miatt van:*

- *Minden lépésnél az $u(t)$ megoldásfüggvényt az elsőfokú $P_1(t)$ polinommal approximáljuk,*
- *$P_1(t)$ polinomban az α együtthatót (azaz a megoldásfüggvény deriváltjait) csak közelítőleg tudjuk meghatározni.*

Mivel az α irányt a megoldásfüggvény t_i és t_{i+1} pontbeli érintőinek iránya határozza meg, ezért általában úgy választjuk meg, hogy ezen két érték közé essék. Ezért a θ paramétert csak a $[0;1]$ intervallumból szokásos megválasztani. A továbbiakban kettő, speciálisan megválasztott $\theta \in [0,1]$ értékhez tartozó numerikus módszert vizsgálunk meg. $\diamond$

2.2.2. Az explicit Euler-módszer

Tekintsük a θ-módszert a $\theta = 0$ megválasztással! Ekkor a (2.8) és a (2.9) a következő numerikus módszert generálják

$$y_{i+1} = y_i + h_i f(t_i, y_i), \quad i = 0,1,\ldots,N-1 \tag{2.10}$$

Mivel y_i az ismeretlen $u(t)$ függvény t_i pontbeli közelítése, ezért értelemszerűen

$$y_0 = u(0) = u_0, \tag{2.11}$$

vagyis az iterációban az $i = 0$ értékhez tartozó y_0 adott érték.

2.2.4. Definíció. *A fenti* (2.10)-(2.11) *képletekkel definiált egylépéses módszert explicit Euler-módszernek nevezzük.*

Legyen

$$g_i(h) = u(t_i) + h\Phi(h, t_i, u(t_i), u(t_{i+1})) - u(t_{i+1}).$$

Ennek a függvénynek rendje sorfejtéssel a Cauchy-feladat egyenletének felhasználásával a pontos megoldás ismerete nélkül is meghatározható.

2.2.5. Definíció. *A $g_i(h)$ függvényt a* (2.2) *alakú Φ numerikus módszer $t_i \in \omega_h$ pontbeli képlethibájának (más szóval lokális approximációs hibájának) nevezzük.*

2.2.6. Definíció. *Azt mondjuk, hogy a Φ numerikus módszer p-ed rendben konzisztens a $t_i \in \omega_h$ rácspontban, ha*

$$g_i(h) = O(h_{p+1})$$

valamely $p > 0$ állandóval.

A lokális approximációs hiba rendje tehát azt mutatja meg, hogy a pontos megoldás milyen pontossággal elégíti ki a Φ-módszer egyenletét.

2.2.7. Megjegyzés. *Megmutatható, hogy az explicit Euler-módszer elsőrendben konzisztens.*$\diamond$

2.2.8. Tétel. *Legyen $t^* \in [0, T]$ tetszőleges rögzített pont. Tekintsük $h \to 0$ mellett a $[0, t^*]$ intervallumon az*

$$\omega_h := \{t_i = ih; \quad i = 0,1,\ldots,n; \quad h = t^*/n\} \tag{2.12}$$

ekvidisztáns rácshálók sorozatát. Ekkor a t^* pontban az explicit Euler módszer elsőrendben konvergens.

2.2.3. Az implicit Euler módszer

A $\theta = 1$ megválasztással a θ-módszer alapján előállíthatjuk a következő numerikus módszert:

$$\begin{aligned} &y_{i+1} = y_i + h_i f(t_{i+1}, y_{i+1}), \quad i = 0,1,\ldots,N-1, \\ &y_0 = u_0. \end{aligned} \tag{2.13}$$

2.2.9. Definíció. *A* (2.13) *képletekkel definiált egylépéses módszert implicit Euler-módszernek nevezzük.*

2.2.10. Megjegyzés. *A implicit Euler-módszert azért nevezzük implicitnek, mert az időben való előrehaladáshoz y_i ismeretében y_{i+1} értékét minden egyes időlépésben egy (tipikusan nemlineáris) egyenlet megoldásával tudjuk csak meghatározni.* $\diamond$

2.2.11. Megjegyzés. *Erről a módszerről is megmutatható, hogy konzisztens és konvergens.* $\diamond$

2.2.4. Runge-Kutta módszerek

A (2.1) Cauchy-feladatra kettőnél magasabb rendű numerikus módszerek megalkotása a korábban ismertetett egylépéses módszerek segítségével problémás. Az egyszerűbb módszerek (explicit Euler-módszer, implicit Euler-módszer, Crank-Nicolson-módszer) legfeljebb másodrendűek, a Taylor-módszerek viszont egy meglehetősen bonyolult előzetes analízist (a parciális deriváltak meghatározását) és azok kiértékelését igénylik. Ebben a részben megmutatjuk, hogy a parciális deriváltak kiszámításának feladata szerencsére megkerülhető.

Másodrendű Runge-Kutta módszerek

Induljunk ki a (2.1) Cauchy-feladatból. A módszer bevezetéséhez először meghatározunk egy másodrendű kétlépéses módszert. Az $u(t)$ megoldást a $t = t^* + h$ pontban Taylor-sorba fejtve nyerjük, hogy:

$$u(t^* + h) = u(t^*) + hu'(t^*) + \frac{h^2}{2!}u''(t^*) + O(h^3). \tag{2.14}$$

A láncszabály alkalmazásával rendre deriválva a (2.1) azonosságot a $t^* \in [0, T]$ pontban, a következő egyenlőségeket nyerjük:

$$\begin{aligned}
&u'(t^*) = f(t^*, u(t^*)),\\
&u''(t^*) = \partial_1 f(t^*, u(t^*)) + \partial_2 f(t^*, u(t^*))u'(t^*),\\
&u'''(t^*) = \partial_{11} f(t^*, u(t^*)) + 2\partial_{12} f(t^*, u(t^*))u'(t^*) + \partial_{22} f(t^*, u(t^*))(u'(t^*))^2 + \partial_2 f(t^*, u(t^*))u''(t^*)
\end{aligned} \tag{2.15}$$

Vezessük be a következő jelöléseket:

$$f = f(t^*, u(t^*)), \quad \partial_i f = \partial_i f(t^*, u(t^*)), \quad \partial_{ij} f = \partial_{ij} f(t^*, u(t^*)), \text{ stb.}$$

Az előbbi egyenlőtlenségekkel és a jelölésekkel átírhatjuk a (2.14) egyenletet a következőképpen:

$$u(t^*+h) = u(t^*)+hf+\frac{h^2}{2!}(\partial_1 f+f\partial_2 f)+O(h^3) = u(t^*)+\frac{h}{2}f+\frac{h}{2}[f+h\partial_1 f+hf\partial_2 f]+O(h^3) \tag{2.16}$$

Mivel tetszőleges $c_1, c_2 \in \mathbb{R}$ esetén a kétváltozós $f : \mathbb{Q} \to \mathbb{R}$ függvény (t, u) pont körüli elsőfokú Taylor-sorba fejtése $f(t + c_1h, u + c_2h) = f(t, u) + c_1h\partial_1 f(t, u) + c_2h\partial_2 f(t, u) + O(h^2)$ alakú, vagyis jelen esetben

$$f(t^* + h, u(t^*) + hf(t^*, u(t^*)) = f + h\partial_1 f + hf\partial_2 f + O(h^2) \tag{2.17}$$

ezért folytathatjuk az átírást a következő alakra:

$$u(t^* + h) = u(t^*) + \frac{h}{2}f + \frac{h}{2}(f(t^* + h, u(t^*) + hf(t^*, u(t^*))) + O(h^3) \tag{2.18}$$

2.2.12. Megjegyzés. *A Runge-Kutta módszereket általában Butcher-táblázat segítségével szokták leírni, mely az alább látható módon néz ki:*

2.2.13. Definíció. *Egy explicit Runge-Kutta típusú módszer*

$$\begin{array}{c|c} a & B \\ \hline & \sigma^T \end{array}$$

alakban felírt paramétereinek táblázatát Butcher-táblázatnak nevezzük.

2.2.14. Tétel. *A* (2.2.13) *Butcher-táblázatú explicit Runge-Kutta típusú módszer pontosan akkor konzisztens, amikor teljesülnek a*

$$Be = a; \quad \sigma^T \cdot e = 1 \tag{2.19}$$

feltételek, azaz

$$\sum_{k=1}^{m} b_{ik} = a_i \text{ minden } i = 1{,}2\cdots, m \text{ esetén és emellett } \sum_{k=1}^{m} \sigma_k = 1. \tag{2.20}$$

Mivel

$$u(t^* + h) - u(t^*) - \frac{h}{2} f - \frac{h}{2}(f(t^* + h, u(t^*) + hf(t^*, u(t^*))) = O(h^3) \tag{2.21}$$

ezért a módszer másodrendű $\diamond$

2.2.15. Megjegyzés. *A* $\sigma = 0.5$ *értékhez tartozó RK2 módszer a* ***Heun-módszert*** *eredményezi. Érdekes megválasztás a* $\sigma = 1$. *Ekkor* $\sigma_1 = 0, \sigma_2 = 1$ *és* $a_2 = b_{21} = 0.5$ *és így a származtatott numerikus módszer*

$$k_1 = f(t_i, y_i), \quad k_2 = f(t_i + 0.5h, y_i + 0.5hk_1), \quad y_{i+1} = y_i + hk_2 \tag{2.22}$$

A (2.22) *másodrendű módszert* ***javított explicit Euler-módszernek*** *nevezzük.* $\diamond$

Célunk érdekében általánosítsuk a (2.18) egyenletet:

$$u(t^* + h) = u(t^*) + \sigma_1 hf(t^*, u(t^*)) + \sigma_2 hf(t^* + a_2 h, u(t^*) + b_{21} hf(t^*, u(t^*))) + O(h^3) \tag{2.23}$$

ahol σ_1, σ_2, a_2 és b_{21} egyelőre tetszőleges paraméterek.

2.2.16. Megjegyzés. *Ha a* $t = t_i$ *pontban felírjuk a* (2.23) *egyenletet, akkor az*

$$y_{i+1} = y_i + \sigma_1 hf(t_i, y_i) + \sigma_2 hf(t_i + a_2 h, y_i + b_{21} hf(t_i, y_i)) \tag{2.24}$$

egylépéses numerikus módszert kapjuk. $\diamond$

A (2.23)-os egyenlet jobb oldalát Taylor-sorba fejtve a

$$\begin{aligned} &u(t^* + h) = u(t^*) + \sigma_1 hf + \sigma_2 h[f + a_2 h\partial_1 f + b_{21} hf\partial_2 f + O(h^3) = \\ &u(t^*) + (\sigma_1 + \sigma_2)hf + h^2[a_2\sigma_2\partial_1 f + \sigma_2 b_{21} f\partial_2 f] + O(h^3) \end{aligned} \tag{2.25}$$

egyenlőséget kapjuk. A (2.2.12) megjegyzés alapján, a (2.16) és a (2.25) képletek összevetéséből azt kapjuk, hogy a (2.24) által meghatározott numerikus módszer pontosan akkor másodrendű, amikor

$$\begin{aligned} \sigma_1 + \sigma_2 &= 1 \\ a_2\sigma_2 &= 0.5 \\ b_{21}\sigma_2 &= 0.5. \end{aligned} \tag{2.26}$$

2.2.17. Tétel. *Tegyük fel, hogy a σ_1, σ_2, a_2 és b_{21} paraméterek megoldásai a* (2.26) *egyenletnek. Ekkor a*

$$k_1 = f(t_i, y_i), \quad k_2 = f(t_i + a_2 h, y_i + h b_{21} k_1) \tag{2.27}$$

$$y_{i+1} = y_i + h(\sigma_1 k_1 + \sigma_2 k_2) \tag{2.28}$$

képletekkel defniált egylépéses explicit numerikus módszer másodrendű.

A fenti tétel feltételeit kielégítő módszert *másodrendű Runge-Kutta módszernek* nevezzük, (a továbbiakben **RK2**).
Észrevehető, hogy a (2.26) egyenletrendszer négy ismeretlenjére csupán három egyenletünk van, ezért a megoldás nem egyértelmű, de szerencsére könnyen belátható, hogy tetszőleges $\sigma \neq 0$ esetén az egyenletrendszerünk megoldásai a következők:

$$\sigma_2 = \sigma, \quad \sigma_1 = 1 - \sigma, \quad a_2 = b_{21} = 0.5\sigma. \tag{2.29}$$

Ennek alapján az RK2 módszerek egy egyparaméteres módszercsaládot alkotnak.

2.2.18. Megjegyzés. *Érdekes tény, hogy a σ tetszőleges paramétert nem lehet már úgy se megválasztani, hogy RK2 másod helyett harmadrendű legyen. Erre bizonyíték a következő példa. Legyen a kiindulási* (2.1) *Cauchy-feladatban $f(t,u) = u$, azaz $u'(t) = u(t)$, melynek deriváltja $u''(t) = u'(t)$. Így hát $u''(t) = u'(t) = u(t)$. Másfelől az f függvény definíciója miatt az $f(t_i, y_i) = y_i$. Erre alkalmazva a* (2.24)*-ös algoritmust*

$$\begin{aligned} y_{i+1} = y_i + \sigma_1 h y_i + \sigma_2 h(y_i + b_{21} h f(t_i, y_i)) = y_i + \sigma_1 h y_i + \sigma_2 h(y_i + b_{21} h y_i) = \\ y_i + h y_i[\sigma_1 + \sigma_2 + h\sigma_2 b_{21}] = y_i[1 + (\sigma_1 + \sigma_2)h + \sigma_2 b_{21} h^2] \end{aligned} \tag{2.30}$$

egyenletet kapjuk. Helyettesítsük be a (2.29)*-es értékeket, így az RK2 módszer erre a feladatra az*

$$y_{i+1} = y_i(1 + h + \frac{h^2}{2}) \tag{2.31}$$

algoritmust adja, ami független a σ paramétertől. Az $u(t)$ pontos megoldás behelyettesítésével az előbbi képletbe megkaphatjuk a lokális approximácós hibát:

$$g_i = u(t_{i+1}) - u(t_i)(1 + h + \frac{h^2}{2}). \tag{2.32}$$

Az $u(t_{i+1})$ kifejezés $t = t_i$ pontbeli sorbafejtése a deriváltakra vonatkozó $u''(t_i) = u'(t_i) = u(t_i)$ egyenlőség következtében $u(t_{i+1}) = u(t_i)(1+h+\frac{h^2}{2}) + O(h^3)$. Ezért tehát $g_i = O(h^3)$ σ tetszőleges megválasztása mellett, azaz minden RK2 módszer legfeljebb másodrendű erre a feladatra. $\diamond$

Magasabb rendű Runge-Kutta módszerek

Ebben a fejezetben magasabbrendben szeretnénk pontos formulákat előállítani kiindulásként a (2.24)-as általános algoritmust véve. Erre azért van szükség, mert gyakorlatban az első- és másodrendű módszerek túl sok időt vennének igénybe a megfelelő pontossághoz.
Induljunk ki a (2.27)-(2.28) által felírt módszerből. Ez a módszer másodrendben pontos. Ahhoz, hogy ez magasabb rendben, azaz jelen esetben harmadrendben is pontos legyen új változókat kell bevezetni és azokat megfelelően kell megválasztani. Tehát a (2.27)-(2.28) alakból kiindulva adódik a következő általánosítás:

$$\begin{aligned} k_1 &= f(t_i, y_i), \\ k_2 &= f(t_i + a_2 h, y_i + h b_{21} k_1), \\ k_3 &= f(t_i + a_3 h, y_i + h b_{31} k_1 + h b_{32} k_2) \end{aligned} \tag{2.33}$$

Ezek alapján az új érték:

$$y_{i+1} = y_i + h(\sigma_1 k_1 + \sigma_2 k_2 + \sigma_3 k_3) \tag{2.34}$$

Későbbi gyakori használata miatt itt vezessük be a következő fogalmat:

2.2.19. Definíció. *Az ún. lokális approximációs hiba, ami a Taylor-sor Taylor-polinommal való helyettesítéséből ered, feltételezve, hogy a t_i pontbeli értéket pontosan ismerjük. Ennek a $[t_i, t_i + h_i]$ intervallum hossza szerinti rendjét, vagyis az $u(t_{i+1}) - T_{n,u}(t_{i+1})$ eltérés h_i szerinti rendjét lokális hibarendnek nevezzük. (Megfelelően sima függvények esetén ez a rend $O(h_i^{p+1})$.)*

A (2.34)-es formulában definiált módszer harmadrendűségéhez a lokális approximációs hiba vizsgálata szükséges. Nem nehéz, de hosszú számolás után megkaphatjuk, hogy a paraméterekre a következő kikötéseket kell tennünk.:

$$\begin{aligned} &a_2 = b_{21}, \quad a_3 = b_{31} + b_{32}, \\ &a_3(a_3 - a_2) - b_{32} a_2 (2 - 3a_2) = 0, \quad \sigma_3 b_{32} a_2 = \frac{1}{6}, \quad \sigma_2 a_2 + \sigma_3 a_3 = \frac{1}{2}, \\ &\sigma_1 + \sigma_2 + \sigma_3 = 1 \end{aligned} \tag{2.35}$$

Így hat egyenletet kapunk nyolc ismeretlenre. A lehetőségek közül kettőt mutatunk be.

1. Legyen
$$\begin{aligned} &a_2 = b_{21} = \frac{1}{3}, \quad a_3 = b_{32} = \frac{3}{2}, \quad b_{31} = 0 \\ &\sigma_1 = \frac{1}{4}, \quad \sigma_2 = 0, \quad \sigma_3 = \frac{3}{4} \end{aligned} \tag{2.36}$$
Az alábbi értékekkel bíró módszer gyakran szerepel alkalmazásokban.

2. A másik harmadrendű módszer:
$$\begin{aligned} &a_2 = b_{21} = \frac{1}{2}, \quad a_3 = 1, \quad b_{31} = -1, \quad b_{32} = 2, \\ &\sigma_1 = \sigma_3 = \frac{1}{6}, \quad \sigma_2 = \frac{2}{3} \end{aligned} \tag{2.37}$$
Ez a módszert leginkább akkor érdemes használni, amikor $\partial_2 f$ közel van a nullához, ugyanis $f(t, u) = f(t)$ esetén ez $O(h^5)$ rendben pontos.

Magasabb pontosság ($p > 3$) eléréséhez általánosítsuk az eddigi eredményeinket! Legyen $m \geq 1$ egy adott egész szám. Definiáljuk a következő, ún. *m-lépcsős explicit Runge-Kutta típusú módszert*:

$$\begin{aligned} k_1 &= f(t_i, y_i), \\ k_2 &= f(t_i + a_2 h, y_i + h b_{21} k_1), \\ k_3 &= f(t_i + a_3 h, y_i + h b_{31} k_1 + h b_{32} k_2), \\ &\vdots \\ k_m &= f(t_i + a_m h, y_i + h b_{m1} k_1 + h b_{m2} k_2 + \cdots + h b_{m,m-1} k_{m-1}) \end{aligned} \tag{2.38}$$

$$y_{i+1} = y_i + h(\sigma_1 k_1 + \sigma_2 k_2 + \cdots + \sigma_m k_m) \tag{2.39}$$

A módszer megadása a képletekben lévő változók meghatározásával történik. Ezen magasabb rendű módszerek közül leginkább a negyedrendű Runge-Kutta módszert szokták alkalmazni. Az algoritmus, azaz a t_{i+1} pontbeli y_{i+1} közelítés meghatározása a már kiszámolt t_i pontbeli y_i közelítésből, a következő: az alábbi képletekkel

$$\begin{aligned} k_1 &= f(t_i, y_i), \\ k_2 &= f(t_i + 0.5h, y_i + 0.5h k_1) \\ k_3 &= f(t_i + 0.5h, y_i + 0.5h k_2) \\ k_4 &= f(t_i + h, y_i + h k_3) \end{aligned} \tag{2.40}$$

kiszámoljuk k_1, k_2, k_3, k_4 értékeket. És az

$$y_{i+1} = y_i + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4) \tag{2.41}$$

képlettel meghatározzuk az új közelítést.

2.2.20. Megjegyzés. *Az explicit Runge-Kutta módszerek konzisztenciájával foglalkoztunk, ám a gyakorlatban sokkal fontosabb a konvergencia. Bebizonyítható, hogy a p-ed rendű konzisztencia mellett a p-ed rendű konvergenciával is rendelkeznek ezek a módszerek. Ez az ún. zéro-stabilitás egyik tulajdonsága. Ennek részletezése azonban meghaladja ezen dolgozat kereteit.* ⋄

Implicit Runge-Kutta módszerek

Megfigyelhető, hogy az általános Runge-Kutta módszerek a következőkkel is leírhatók. Legyen $B \in \mathbb{R}^{m \times m}$ egy tetszőleges mátrix, $\sigma \in \mathbb{R}^m$ egy adott vektor, és $a = Be$. Ha B nem szigorúan alsó háromszög mátrix, akkor *implicit Runge-Kutta típusú módszernek* (IRK) nevezzük. Ha B alsó (de nem szigorúan alsó) háromszögmátrix, akkor a módszert *diagonálisan implicit Runge-Kutta típusú módszernek* (DIRK) nevezzük.
Ezekben a módszerekben a fenti Runge-Kutta módszerektől eltérően a k_i értékek kiszámolásához egy (általában nemlineáris) egyenletet, illetve az implicit Runge-Kutta módszer esetében m egyenletből álló egyenletrendszer megoldásait kell megtalálni. Habár ez a módszer bonyolultságát fokozza az explicithez képest, mégis a gyakorlatban gyakrabban alkalmazott módszerekről

beszélünk. Ennek egyik oka, hogy ugyanazon lépésszám mellet magasabb a konzisztencia rendje (és így a konvergencia is), másik oka, hogy az explicit Runge-Kutta módszerektől eltérően a magasabb rendben pontos módszerek is jó kvalitatív (mint az A-stabilitás) tulajdonságokkal rendelkeznek.

1. Ilyen egylépcsős diagonálisan implicit Runge-Kutta típusú módszer a következő:

$$\begin{aligned} k_1 &= f(t_i + h, y_i + hk_1) \\ y_{i+1} &= y_i + hk_1 \end{aligned} \tag{2.42}$$

Tehát az algoritmus a következőképp néz ki: Első lépésben megoldjuk a k_1 ismeretlenre az első egyenletet (általában Newton-féle iterációval), majd a megoldást behelyettesítjük a második képletbe. Határozzuk meg a módszer Φ függvényét! Mivel a második képletből $hk_1 = y_{i+1} - y_i$, ezt behelyettesítve az első egyenletbe $k_1 = f(t_i + h, y_i + (y_{i+1} - y_i)) = f(t_i + h, y_{i+1})$. Ezért, ismét a második összefüggésből, $y_{i+1} = y_i + hf(t_i + h, y_{i+1})$, azaz $\Phi(h, t_i, y_i, y_{i+1}) = f(t_i + h, y_{i+1})$. Így ez a módszer elsőrendű.

2. Szintén egy egylépcsős diagonálisan implicit Runge-Kutta típusú módszer, amely a következőt jelenti:

$$\begin{aligned} k_1 &= f(t_i + 0.5h, y_i + 0.5hk_1) \\ y_{i+1} &= y_i + hk_1. \end{aligned} \tag{2.43}$$

Az algoritmus lépései itt is hasonlóak mint az előbb. Itt is a Φ függvény kiszámolásával kezdjük. A második képletből $hk_1 = y_{i+1} - y_i$. Behelyettesítve az első egyenletbe: $k_1 = f(t_i + 0.5h, y_i + 0.5(y_{i+1} - y_i)) = f(t_i + 0.5h, 0.5(y_i + y_{i+1}))$. Tehát $\Phi(h, t_i, y_i, y_{i+1}) = f(t_i + 0.5h, 0.5(y_i + y_{i+1}))$ és a megfelelő egylépéses módszer

$$y_{i+1} = y_i + hf(t_i + 0.5h, 0.5(y_i + y_{i+1})) \tag{2.44}$$

alakú. A (2.44) diagonálisan implicit Runge-Kutta típusú módszert **implicit középponti szabálynak** nevezzük. A módszer rendjét a

$$g_i = u(t_{i+1}) - u(t_i) - hf(t_i + 0.5h, 0.5(u(t_i) + u(t_{i+1})))$$

kifejezés nagyságrendjének meghatározásával nyerjük. A szokásos sorbafejtéssel a következőket kapjuk:

$$\begin{aligned} u(t_{i+1}) - u(t_i) &= hu'(t_i) + \frac{h^2}{2}u''(t_i) + O(h^3) \\ f(t_i + 0.5h, 0.5(u(t_i) + u(t_{i+1}))) &= f(t_i + 0.5h, u(t_i) + 0.5hu'(t_i) + O(h^2)) \\ &= f(t_i, u(t_i)) + 0.5h\partial_1 f(t_i, u(t_i)) + 0.5hu'(t_i)\partial_2 f(t_i, u(t_i)) + O(h^2) \\ &= f(t_i, u(t_i)) + 0.5h[\partial_1 f(t_i, u(t_i)) + f(t_i, u(t_i))\partial_2 f(t_i, u(t_i))] + O(h^2). \end{aligned}$$

Behelyettesítve ezen értékeket a g_i kifejezésbe és felhasználva a (2.43)-as formulából a második összefüggést, megkapjuk, hogy $g_i = O(h^3)$, vagyis az implicit középponti szabály másodrendű.

3. A következő példában egy kétlépcsős implicit Runge-Kutta módszer szerepel:

$$\begin{aligned} k_1 &= f(t_i, y_i) \\ k_2 &= f(t_i + h, y_i + 0.5k_1 + 0.5k_2) \\ y_{i+1} &= y_i + 0.5hk_1 + 0.5hk_2. \end{aligned} \tag{2.45}$$

A harmadik képletből következik, hogy $0.5h(k_1 + k_2) = y_{i+1} - y_i$. Ez és az első egyenlet alapján a második egyenlet: $k_2 = f(t_i+h, y_i+(y_{i+1}-y_i)) = f(t_i+h, y_{i+1})$. Ezt a k_2 értéket és a k_1 értéket behelyettesítve a harmadik egyenletbe megkapjuk az eddig is keresett Φ függvényt, vagyis $\Phi(h, t_i, y_i, y_{i+1}) = 0.5[f(t_i, y_i) + f(t_i + h, y_{i+1})]$ Vagyis megkaptuk a **Crank-Nicolson-módszert**.

Ha a fenti egyenletek pontosságára vagyunk kíváncsiak, megkapjuk, hogy a módszerek pontossága (p) kedvezőbb, mint a lépcsőszám (m). Például az egylépcsős trapézszabály másodrendű, a kétlépcsős Gauss-féle alappontokkal rendelkező módszer negyedrendű, a kétlépcsős Radau-módszer harmadrendű stb. Tehát az explicit Runge-Kutta típusú módszertől eltérően, a $p > m$ eset is lehetséges. (ez valószínűleg azért van így, mert az implicit Runge-Kutta típusú módszernél több a szabadon megválasztható paraméterünk.) Az is belátható, hogy adott lépcsőszám esetén $p \leq 2m$.

2.2.21. Megjegyzés. *A módszereket, ahol $p = 2m$ maximális pontosságú implicit Runge-Kutta módszernek nevezzük. Ilyen például az implicit Euler-módszer, az implicit középponti szabály.* $\diamond$

3. fejezet

Modellek

Ebben a fejezetben különböző példákon bemutatjuk, hogy az előző fejezetben leírt numerikus módszerek milyen esetekben, milyen hatékonysággal működnek.
A már eddig is megszokott módon az elsőrendű differenciálegyenleteket vizsgáljuk, azaz az $u(t) = f(t, u(t))$ egyenletet az $u(0) = u_0$ kezdeti feltétellel. A numerikus megoldás az ismeretlen u(t) függvény egy $t_i = ih \quad (i = 0,1,\cdots,N)$ rácshálón való közelítését jelenti, ahol az $u(t_i)$ közelítését jelentő y_i értékét valamilyen képlet segítségével határozzuk meg.
A módszerek pontosságát a lokális approximációs hiba jellemzi, amely azt fejezi ki, hogy a pontos megoldás rácshálón vett vetülete h milyen rendjében elégíti ki a numerikus megoldást meghatározó sémát.
A példákat az alábbi, már korábban begmutatott egylépéses módszerekkel fogjuk megoldani:

- Explicit Euler-módszer: $y_i = y_{i-1} + f(t_{i-1}, y_{i-1})$.

- Implicit Euler-módszer: $y_i = y_{i-1} + f(t_i, y_i)$.

Ezek elsőrendűek és általánosításuk a θ-módszer. Az utóbb említett módszer, ahogy a nevében is szerepel implicit, vagyis csak egy egyenlet megoldásával kaphatjuk meg y_i-t. Ezt elkerülendő módosíthatjuk az egyenleteinket, -ezzel explicitté téve őket,- és így megkaphatjuk a javított-Euler módszert, illetve általánosításként megkaphatjuk a Runge-Kutta módszereket. Utóbbiak több köztes érték segítségével számolják ki az y_{i-1} értékből az y_i értékét. Ezek a módszerek a köztes értékek számától (az ún. lépcsőszámtól) függően általában magasabb rendben pontosak.
Az egylépéses módszerek általánosítása a lineáris többlépéses módszerek, amelyek alakja

$$a_0 y_i + a_1 y_{i-1} + \cdots + a_m y_{i-m} = h(b_0 f_i + b_1 f_{i-1} + \cdots + b_m f_{i-m}), \quad i = m, m+1, \cdots,$$

ahol, $f_i = f(t_i, y_i)$, és a_k és b_k a módszert definiáló adott paraméterek. Ezek a módszerek b_0 értékétől függően szintén lehetnek explicitek ($b_0 = 0$) és implicitek ($b_0 = 0$). Pontosságukat az m lépcsőszám határozza meg.

3.1. Numerikus modellek Matlabban

Ebben a részben bemutatjuk a fent leírt módszerek numerikus megoldását Matlab programnyelv segítségével különböző feladatokra.

3.1.1. Euler-módszerek

Explicit Euler-módszer

Itt az általános explicit Euler módszer MATLAB-beli kódja látható, h a lépésfinomság, $xinit$ a tartomány kezdete, $xfinal$ a tartomány vége, n a lépésköz, $yinit$ a kezdeti feltétel. Ez az algoritmus megadja az egyenlet megoldását és a lépésfinomságból adodó hibát is. A dolgozat későbbi részeiben nézünk konkrét példákat. Az explicit Euler-módszert leíró egyenlet:

$$y_{n+1} = y_n + hf(t_n, y_n)$$

Ennek az algoritmusnak a MATLAB kódja:

```
function [x,y,hiba]=euler_forward(f,xinit,yinit,xfinal,n)
% f : a feladatunk,
% n : n reszre osztjuk az idointervallumunkat,
% h : [xinit,xfinal]-t n reszre osztjuk, azaz az algoritmus
%     lepesfinomsaga,
% xinit : [xinit,xfinal] az idointervallumunk kezdete,
% xfinal : [xinit,xfinal] az idointervallumunk vege
% hiba: a modszer hibaja
 h=(xfinal-xinit)/n;
 x=[xinit zeros(1,n)];
 y=[yinit zeros(1,n)];
 hiba=zeros(1,n+1);

 for i=1:n
     x(i+1)=x(i)+h;
     y(i+1)=y(i)+h*f(x(i),y(i));
    hiba(i+1)=(y(i+1)-y(i))/h-f(x(i),y(i));
 end

 end
```

Implicit Euler-módszer

Az implicit Euler-módszer algoritmusa a következő:

$$y_{n+1} = y_n + hf(t_{n+1}, y_{n+1})$$

Látható, hogy az egyenletben y_{n+1} részben önmaga egy függvényével van deklarálva, ezért a módszer használatához először a feladatot át kell alakítani, ebből adódóan a feladatot általában Newton- vagy fixpont-iterációval oldjuk meg.

3.1.2. Runge-Kutta módszerek

Ebben a részben a Runge-Kutta módszereknek két fajtáját mutatjuk be. A másod- illetve a negyedrendű Runge-Kutta módszereket.

Másodrendű Runge-Kutta módszer

Léteznek a MATLAB programrendszerben beépített programok, amelyek alkalmasak akár nagy pontossággal és hatékonyan megoldani a kezdetiérték-feladatokat. Ilyen például az 'ode23' Runge-Kutta típusú módszer, amelyet Bogacki-Shampine-módszernek is nevezünk. Ez egy explicit $(2, 3)$-típusú Runge-Kutta-módszer. Igazából akkor hatékony, amikor olcsón szeretnénk kevésbé pontos megoldást kapni. (Általában nem merev, vagy csak nagyon kis merevségű feladatokra alkalmazzuk.) Ennek meghívása a '[T1, Y23] = ode23(f,[kezdőidő,végidő])' utasítás. Itt is két kimenő paraméter van, az idő és a közelítő értékek vektora. A bemenő paraméterek rendre: az alfüggvény, ami leírja a differenciálegyenletünket, az idővektor, hogy melyik időpontokban számoljuk ki a megoldást, és a kezdeti-érték vektor.
Ezenkívül a következő egy általunk megalkotott algoritmus, amit korábban Javított Euler-módszerként mutattunk be:

```
function [x,y,h,hiba]=euler_javitott(f,xinit,yinit,xfinal,n)

% f : a feladatunk,
% n : n reszre osztjuk az idointervallumunkat,
% h : [xinit,xfinal]-t n reszre osztjuk, azaz az algoritmus
%     lepesfinomsaga,
% xinit : [xinit,xfinal] az idointervallumunk kezdete,
% xfinal : [xinit,xfinal] az idointervallumunk vege
% hiba: a modszer hibaja

 h=(xfinal-xinit)/n;
 x=[xinit zeros(1,n)];
 y=[yinit zeros(1,n)];
 hiba=zeros(1,n+1);

 for i=1:n
     x(i+1)=x(i)+h;
```

```
    y(i+1)=y(i)+h*(f(x(i)+0.5*h,y(i)+0.5*h*f(x(i),y(i))));
   hiba(i+1)=(y(i+1)-y(i))/h-f(x(i),y(i));
end

end
```

Negyedrendű Runga-Kutta módszer

Szintén beépített rutin program az 'ode45', ami egy ún. beágyazott Dormand-Prince-módszer. Ez egy egylépéses, váltakozó lépésközű negyed-, ötödrendű Runge-Kutta módszer. A beépített függvényen kívül, pedig bemutatjuk a korábban definiált negyedrendű Runge-Kutta módszer algoritmusát:

```
function [x,y,h,hiba]=RK4(f,xinit,yinit,xfinal,n)
 h=(xfinal-xinit)/n;
 x=[xinit zeros(1,n)]; y=[yinit zeros(1,n)];
 hiba=zeros(1,n+1);
 for i=1:n
     x(i+1)=x(i)+h;
     y(i+1)=y(i)+(h/6)*(f(x(i),y(i))+2*f(x(i)+0.5*h,y(i)+
         0.5*h*f(x(i),y(i)))+2*f(x(i)+0.5*h,y(i)+0.5*h*f(x(i)+
         0.5*h,y(i)+0.5*h*f(x(i),y(i))))+f(x(i)+h,y(i)+h*f(x(i)+
         0.5*h,y(i)+0.5*h*f(x(i)+0.5*h,y(i)+0.5*h*f(x(i),y(i))))));
     hiba(i+1)=(y(i+1)-y(i))/h-f(x(i),y(i));
 end
 end
```

3.2. Egy egyszerű példa

Bevezetésként tekintsük az alábbi [8]-ból származó egyszerűbb feladatot:

$$\begin{aligned} 4\dot{y}(x) &= xy(x) + 2 \\ y(0) &= 3 \end{aligned} \tag{3.1}$$

kezdetiérték-feladatot. Számítsuk ki a megoldás közelítő értékét a $t = 2$ pontban, a $h = \frac{1}{16}, \frac{1}{32}, \frac{1}{64}, \frac{1}{128}, \frac{1}{256}, \frac{1}{512}, \frac{1}{1024}$ lépésközök esetén, ha a módszer:

- explicit Euler,
- implicit Euler,

- javított Euler,
- RK4!

A feladat numerikus megoldása érdekében átírhatjuk így:

$$\begin{aligned} \dot{y} &= \frac{xy+2}{4} \\ y(0) &= 3 \end{aligned} \tag{3.2}$$

A kezdetiértéken nem szükséges változtani.

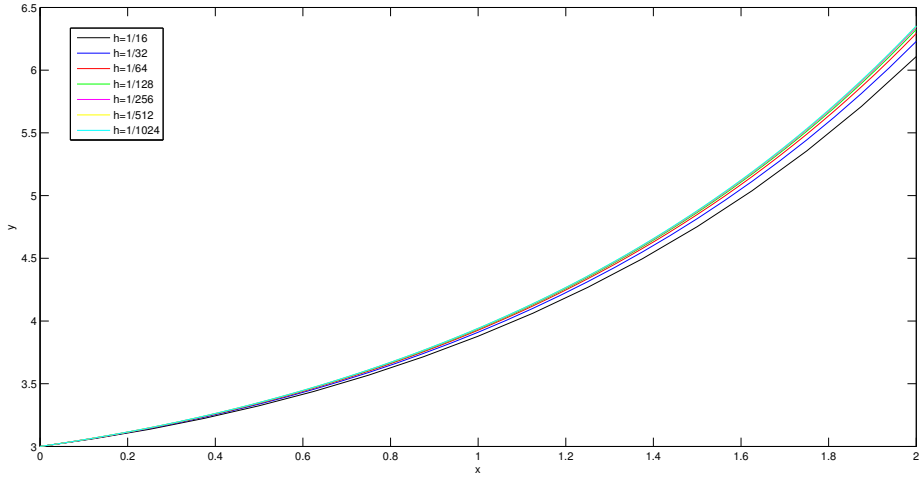
Numarikus megoldás ♦ A feladatot az eddig vett algoritmusokkal oldjuk meg MATLAB segítségével.

- **Explicit Euler-módszer** Az algoritmus MATLAB-beli kódja:

```
xinit=0;                          %A vizsgalt intervallum kezdete
yinit=3;                          %A kezdeti feltetel
f=@(x,y) (x*y+2)/4;               %A feladat
xfinal=2;                         %A vizsgalt intervallum vege

%Kulonbozo nagysagu (2^n) felosztasokhoz:
hatvany=4:1:10;
n=2.^hatvany;
linecolor=['k','b','r','g','m','y','c'];

%A script:
for i=1:length(n)
    [x,y,h,hiba]=euler_forward(f,xinit,yinit,xfinal,n(i));

    %Hibak tablazata:
    tablazat(i,:)=[h,y(end),hiba(end)]
end
```

A 3.1 ábrán tekinthetjük meg a (3.2)-es egyenlet numerikus megoldását explicit Euler-módszerrel különböző lépésfinomsággal.

- **Implicit Euler-módszer** Az egyenletek átalakíthatóak úgy, hogy megoldható legyen az explicit Euler-algoritmushoz hasonló módon. Ez ebben a feladatban a következő módon történik:
  Általánosítva y_n a következőképp néz ki:



3.1. ábra. A (3.2)-es egyenlet numerikus megoldása explicit Euler-módszerrel különböző h lépésfinomsággal.

$$y_n = y_{n-1} + hf(t_n, y_n)$$

Ez a (3.2) esetén:

$$y_n = y_{n-1} + h\frac{x_n y_n + 2}{4}$$
$$y_n = y_{n-1} + \frac{hx_n y_n}{4} + \frac{h}{2}$$
$$y_n(1 - \frac{hx_n}{4}) = y_{n-1} + \frac{h}{2}$$
$$y_n = \frac{y_{n-1} + \frac{h}{2}}{1 - \frac{hx_n}{4}}$$

Ezt az egyenletet már meg tudjuk oldani numerikusan az explicit Euler-módszerhez hasonlóan y_n-ben implicit egyenlet megoldása nélkül.

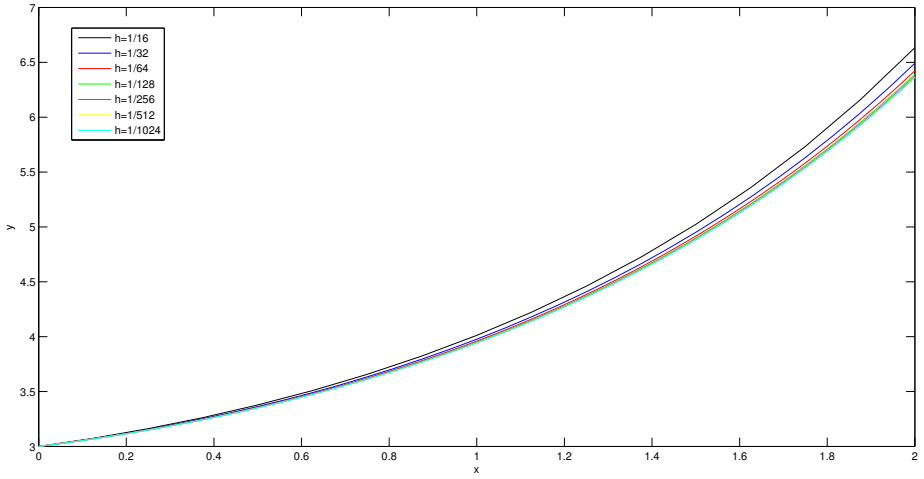
A 3.2 ábra az implicit Euler-módszer különböző lépésfinomságnál.

- **Javított Euler-módszer** Ennél a módszernél nincs más dolgunk, mint a fentebb ismertetett javított Euler-módszert meghívni a feladat paramétereivel

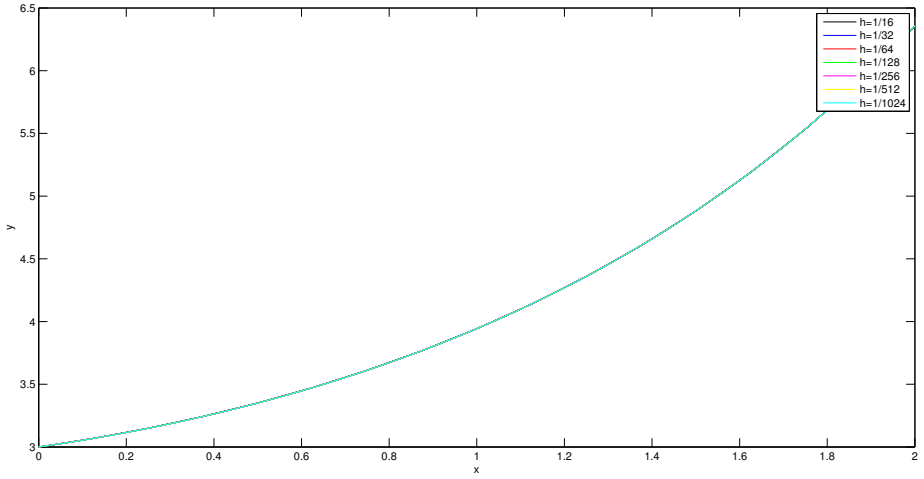
  Eredményeinket a 3.3 ábrán mutatjuk be.

- **Negyedrendű Runge-Kutta módszer** Hasonlóan az explicit, ill. javított Euler-módszerhez meghívhatjuk a fejezet elején ismertetett módszert a feladat adataira.
  A megoldás a 3.4 ábrán látható.

  Az alábbi táblázatban összefoglaljuk a különböző módszerek hibáját, különböző lépésfinomság mellett a (3.2) példán keresztül:



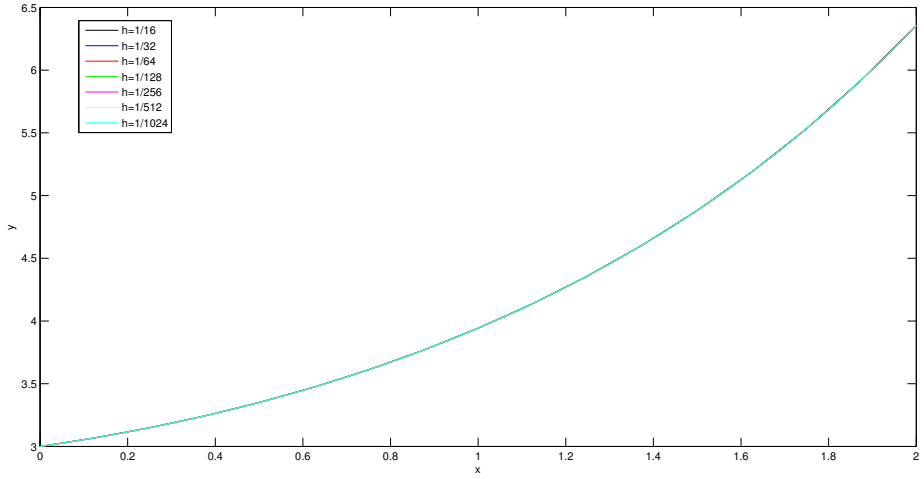
3.2. ábra. A (3.2)-es egyenlet numerikus megoldása implicit Euler-módszerrel különböző h lépésfinomsággal.



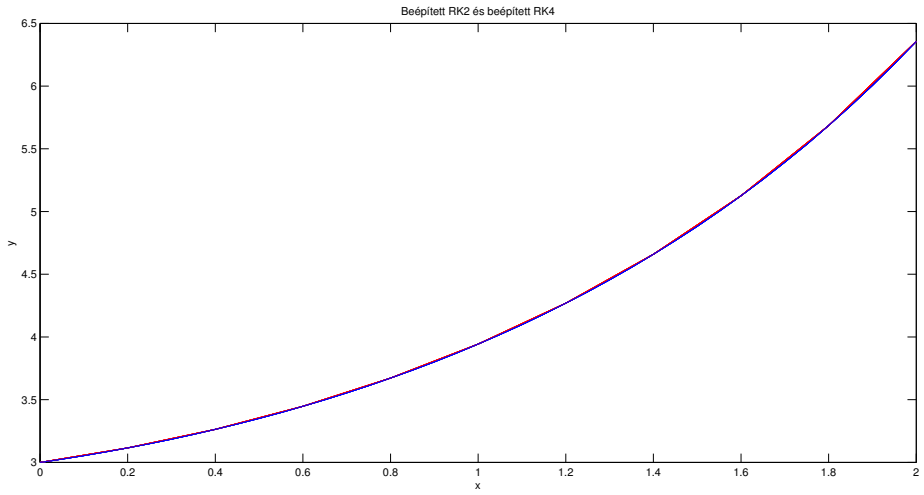
3.3. ábra. A (3.2)-es egyenlet numerikus megoldása javított Euler-módszerrel különböző h lépésfinomsággal.

3.3. Redukált háromtest probléma

Egyik alapfeladata az égi mechanikának meghatározni n számú pontszerű égitest mozgását, ha rájuk csak a Newton-féle kölcsönös gravitációs vonzerő hat. Ehhez vezessünk be a következőket. Legyenek $P_1, \cdots, P_n$ az n db tömegpontunk, melyeknek tömege rendre $m_1, \cdots, m_n$. Jelölje P_i helyvektorát $r_i = (x_i, y_i, z_i)$ az általános O_{xyz} inerciarendszerben. Ekkor a P_i-re a $P_j \quad (i \neq j)$ által kifejtett erő a Newton-féle gravitációs törvény alapján:



3.4. ábra. A (3.2)-es egyenlet numerikus megoldása RK4 különböző h lépésfinomsággal.



3.5. ábra. A (3.2)-es egyenlet numerikus megoldása a beépített RK2 és RK4 módszerekkel $n = 1024$ esetén

$$\underline{F}_{ij} = k^2 \frac{m_i m_j}{r_{ij}^2} \frac{\underline{r}_{ij}}{r_{ij}}$$

ahol $r_{ij} = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2 + (z_j - z_i)^2}$. Mivel a P_i-re ható eredő erőt az összes j-re kell venni, ezért szummázva, és F_i-t kifejtve az n test probléma mozgásgegyenletei:

$$m\ddot{a} = k^2 \sum_{j \to 1}^{n} \frac{m_i m_j}{r_{ij}^3} \underline{r}_{ij}, \tag{3.3}$$

3.1. táblázat. Hibák táblázata, ahol EE=explicit Euler-módszer hibája,IE=Implicit Euler-módszer hibája, JE=Javított Euler-módszer hibája, RK4= negyedrendű Runge-Kutta módszer hibája különböző h lépésfinomság mellett

		EE		IE		JE		RK4	
n	h=2/n	y(n)	hiba	y(n)	hiba	y(n)	hiba	y(n)	hiba
16	0.125	6.1087	-8,88E-16	6,6341	0,4310	6,3524	0,1916	6,3568	0,1968
32	0.0625	6.2295	-1,78E-15	6,4915	0,2148	6,3557	0,1013	6,3568	0,1026
64	0.03125	6.2923	6,22E-15	6,4232	0,1073	6,3566	0,0521	6,3568	0,0524
128	0.0156	6.3244	2,04E-14	6,3898	0,0536	6,3568	0,0264	6,3568	0,0265
256	0.0078	6.3405	-1,73E-14	6,3733	0,0268	6,3568	0,0133	6,3568	0,0133
512	0.0039	6.3487	4,26E-14	6,3650	0,0134	6,3568	0,0067	6,3568	0,0067
1024	0.0020	6.3528	2,13E-14	6,3609	0,0067	6,3568	0,0033	6,3568	0,0033

A matematikai modell ♦ A redukált háromtest probléma az n probléma egy egyszerűsített $n = 3$-ra vett problémája. Ebben a feladatban tömegközépponthoz rögzített forgó koordinátarendszerben a két tömegpontunknak a Napnak (P_1, tömege m_1) és a Jupiternek (P_2, tömege m_2) fix helye van, s a harmadik égitest hozzájuk képesti mozgását tanulmányozzuk. P_1 és P_2 a kölcsönös vonzás következtében körpályán mozognak, a harmadik test tömege elhanyagolható. Ekkor a harmadik test mozgásegyenletei a következők:

$$\begin{aligned} \ddot{x} - 2\dot{y} &= \frac{\partial \Omega}{\partial x} \\ \ddot{y} + 2\dot{x} &= \frac{\partial \Omega}{\partial y}, \end{aligned} \tag{3.4}$$

ahol

$$\Omega = \frac{1}{2}((1-\mu)r_1^2 + \mu r_2^2) + \frac{1-\mu}{r_1} + \frac{\mu}{r_2} \text{ a potenciál}$$
$$r_1 = \sqrt{(x-\mu)^2 + y^2} \text{ a harmadik égitest Naptól}$$
$$r_2 = \sqrt{(x+1-\mu)^2 + y^2} \text{ a Jupitertől mért távolsága}$$
$$\mu = \frac{m_2}{m_1 + m_2} \text{ a tömegparaméter (Nap-Jupiter rendszer esetén } \mu = 9.538752533 * 10^{-4})$$

Ezekkel ekvivalens a következő egyenletrendszer:

$$\begin{aligned} \dot{x} &= u, \quad \dot{u} = 2v + \frac{\partial \Omega}{\partial x} \\ \dot{y} &= v, \quad \dot{v} = -2u + \frac{\partial \Omega}{\partial y} \end{aligned} \tag{3.5}$$

Polárkoordinátákkal:

$$\begin{aligned} x &= r\cos\varphi + \mu - 1, \\ y &= r\sin\varphi, \\ r &= r_2 \end{aligned}$$

Ezzel felírva (3.5):

$$\begin{aligned}
\dot{\varphi} &= \frac{1}{r}(v\cos\varphi - u\sin\varphi),\\
\dot{r} &= u\cos\varphi + v\sin\varphi,\\
\dot{u} &= 2v + (r\cos\varphi + \mu - 1)(1 - \frac{1-\mu}{r_1^3} - \frac{\mu}{r^3}) + \mu(\mu-1)(\frac{1}{r_1^3} - \frac{1}{r^3})\\
\dot{v} &= -2u + r\sin\varphi(1 - \frac{1-\mu}{r_1^3} - \frac{\mu}{r^3}),\\
r_1^2 &= r^2 + 1 - 2r\cos\varphi.
\end{aligned} \tag{3.6}$$

Az alábbi kezdeti feltételekkel megmutatjuk a harmadik test mozgását:

$$r_0 = r(0), \quad u_0 = -\sqrt{\dot{r}_0^2 + r_0^2\dot{\varphi}_0^2}\sin\varphi_0$$
$$\varphi_0 = \varphi(0), \quad v_0 = \sqrt{\dot{r}_0^2 + r_0^2\dot{\varphi}_0^2}\cos\varphi_0$$

A felhasznált paraméterek:

a Gauss-féle gravitációs állandó: $\mathbf{k} = 0.01720209895$,

a Jupiter tömege: $\mathbf{m_J} = 18988 * 10^{23}$ kg,

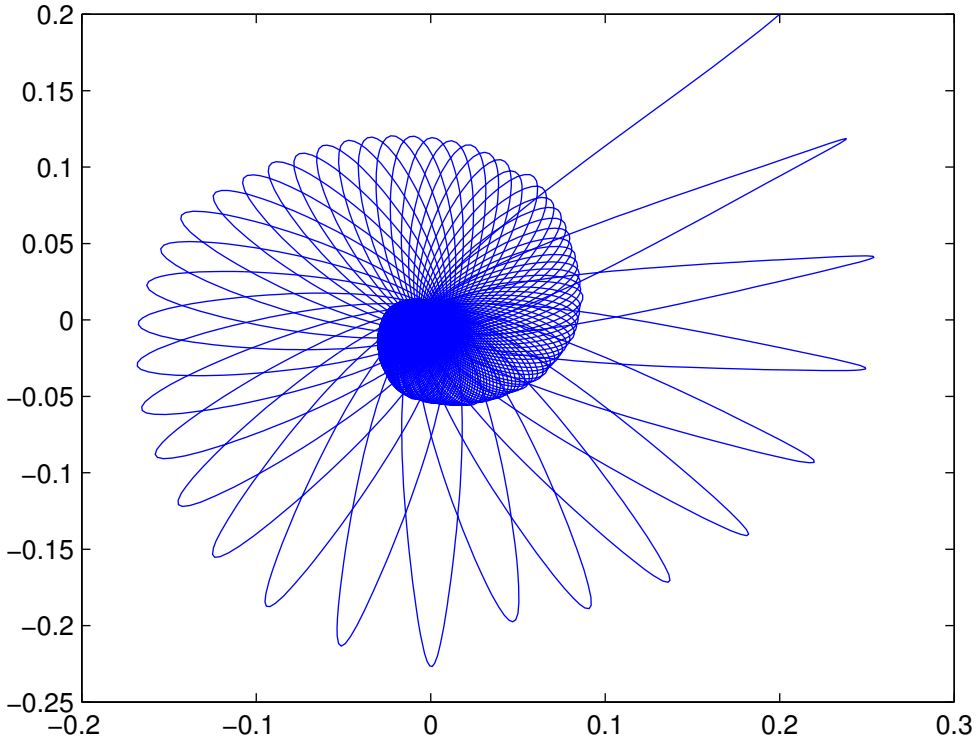
a Nap-Jupiter távolság:$\mathbf{a_J} = 5.202833480999$ cs.e.,

a Jupiter periódusa a Nap körül: $\mathbf{P_J} = 11.857$ év,

a Jupiter sugara: $\mathbf{R_J} = 0.0004772661236$ cs.e..

Numerikus megoldás ♦ Ennek a feladatnak az algoritmusa a következőképpen néz ki:

```
function u = test(t,vv)

m=9.538752533*10^-4;

%A potencial (omega) x szerinti parcialis derivaltja:
f=@(x,y)( (1-m)*(x-m)+m*(x+1-m)-(1-m)*(x-m)/sqrt(((x-m)^2+y^2)^3)-
        m*(x+1-m)/sqrt(((x+1-m)^2+y^2)^3));

%A potencial (omega) y szerinti parcialis derivaltja:
g=@(x,y)( y+m*y-(1-m)*y/sqrt(((x-m)^2+y^2)^3)-
        m*y/sqrt(((x+1-m)^2+y^2)^3));

%A fuggvenyek:
u = zeros(4,1);
u(1,1) = vv(3,1);
u(2,1) = vv(4,1);
u(3,1) = 2*vv(4,1) + f(vv(1,1),vv(2,1));
u(4,1) = -2*vv(3,1) + g(vv(1,1),vv(2,1));
```



3.6. ábra. A harmadik test mozgásegyenlete a korlátozott háromtest problémából

A harmaik test mozgásegyenlete a 3.6 ábrán látható.
A feladat a [4] dokumentum alapján készült.

3.4. Lorenz-rendszer avagy a pillangó-hatás

A feladat részletei az [5] dokumentumból származnak. 1962-ben Lorenz keresett egy egyszerű modellt az időjárás előrejelzésre és egyszerűsíteni a hő átadási egyenleteket a következő három egyenletre:

$$\begin{aligned} \frac{dx}{dt} &= 10(y-x) \\ \frac{dy}{dt} &= -xz + 28x - y \\ \frac{dz}{dt} &= xy - \frac{8}{3}z \end{aligned} \tag{3.7}$$

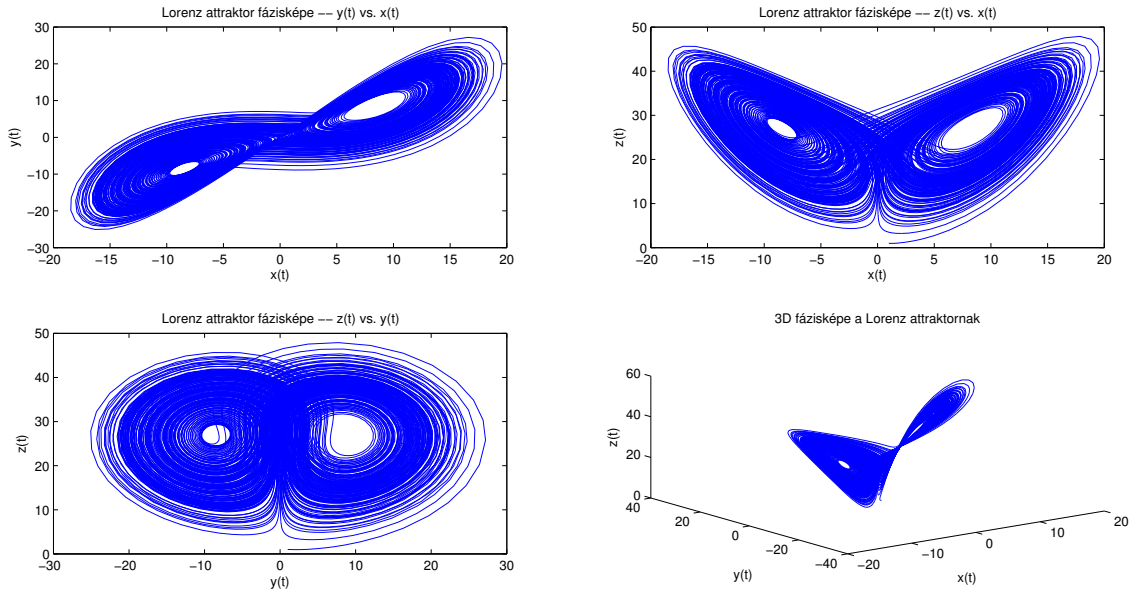
Az egyenletek megoldásai ezenkívül megadják azt a bonyolult viselkedést, ami felkeltette az érdeklődést a káoszról.

Numerikus megoldás ♦ Ennél a feladatnál két különböző fájlban deklaráltuk a feladatot és a paramétereket:

A paraméterek:

```
function dy = lorenz(t,y)
dy = zeros(3,1);

%Parameterek:
P = 10;
r = 28;
b = 8/3;

%Fuggvenyek:
dy(1) = P*(y(2) - y(1));
dy(2) = -y(1)*y(3) + r*y(1) - y(2);
dy(3) = y(1)*y(2) - b*y(3);
```

A feladat megoldása a beépített RK4 módszerrel:
'[t,y] = ode45('lorenz',[0 250], [1.0 1.0 1.0]);'



3.7. ábra. Lorenz rendszer

3.5. Ragadozó-zsákmány modell

A feladat adatai a [7] honlapról származnak. Határozzuk meg a bálnák(W) és krillek(K) arányának 50 időegység alatti változását, ha kezdetben 0.1 bálnára 1 krill jut és tudjuk, hogy a változást leíró egyenletek a következőképpen néznek ki:

$$u = K - 2K^2 - KW$$
$$v = -2W + 6KW.$$

A matematikai modell ♦ Ebben a részben két (vagy több) különböző populáció egymásra gyakorolt hatásával foglalkozunk, vagyis a Lotka-Volterra modellek közül fogjuk vizsgálni az egyiket. Ez a modell a következőképp épül fel: legyen n populáció, $p_i \geq 0$ az i-edik populáció denzitása (vagyis az egyedek területre vonatkoztatott sűrűsége), r_i a belső növekedési rátákat, az adott populáció kvalitatív tulajdonságát határozzák meg. A hatás jelen modell esetében a $p_i e_i j p_j$ (a j-edik populáció i-edikre kifejtett hatása) alakban adható meg. Ekkor a Lotka-Volterra általános differenciálegyenlete:

$$\dot{p_i} = p_i(r_i + \sum_{j=1}^{n} e_{ij} p_j) \tag{3.8}$$

A modellek több típusra is bonthatók. Az eredeti zsákmány-ragadozó modell, mikor az egyik populáció tápláléka a másik populáció (pl. oroszlán-zebra), de a populációkon belül nincs kölcsönhatás (azaz $e_{ii} = 0$). Az alábbi egyenletrendszerben i felel meg a ragadozónak és j a zsákmánynak, így $e_{ij} > 0$ és $e_{ji} < 0$ és a modell a következőképp néz ki:

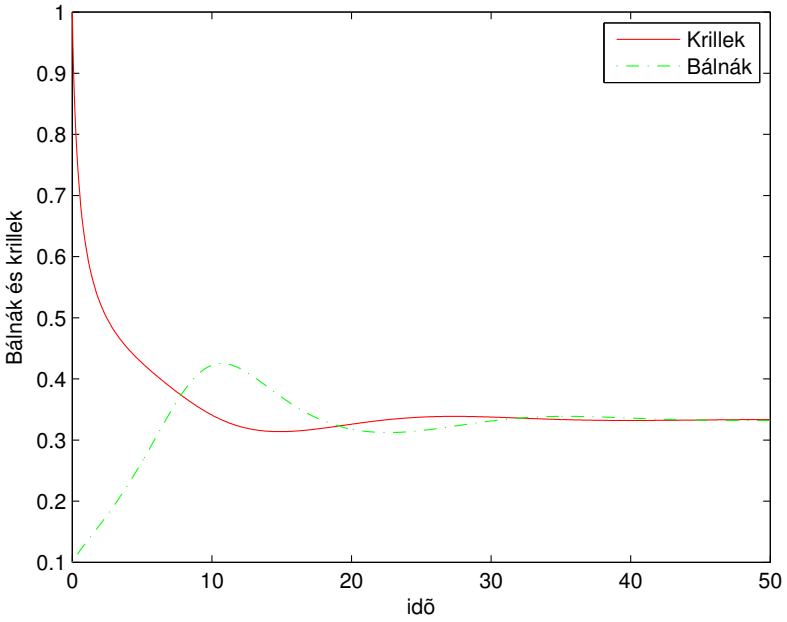
$$\begin{aligned} \dot{p_1} &= p_1(r_1 - |e_{12}|p_2) \\ \dot{p_2} &= -p_2(-r_2 - |e_{21}|p_1) \end{aligned} \tag{3.9}$$

Numerikus megoldás ♦ Ezt a feladatot az explicit Euler-módszerrel oldottam meg, melynek algoritmusa erre a feladatra:

```
%Kezdo ertekek
tinit = 0.0;              % periodus kezdete
tfinal = 50.0;            % perodus vege
K(1)=1;                 % kezdo krill populacio
W(1)=0.1;                 % kezdo balna populacio
u=@(K,W) K-2*K^2-K*W;     %a populacio valtozasat leiro egyenlet
v=@(K,W) -2*W+6*K*W;      %a populacio valtozasat leiro egyenlet
n = 1000;                 % lepesszam
h = (tfinal-tinit)/n; % lepesfinomsag
T=tinit:h:tfinal;     % ido

% Explicit Euler-modszer:
for i = 1:n
  K(i+1) = K(i) + h*K(i)*u(K(i),W(i));
  W(i+1) = W(i) + h*W(i)*v(K(i),W(i));
```

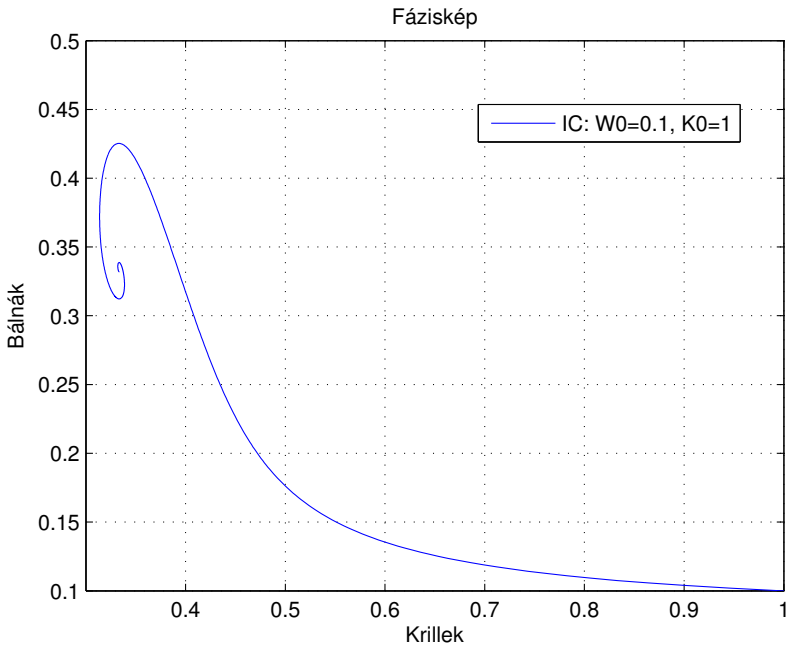
```
end;
```

A populációk változását a 3.8 képen láthatjuk.



3.8. ábra. A populációk változása

Trajektóriák $p_1(0) = 1.0$, $p_2(0) = 0.1$ kezdeti feltétellel.



3.9. ábra. A populációk fázisképe

Irodalomjegyzék

[1] Faragó István, Horváth Róbert: Numerikus módszerek, 2013

[2] Simon Péter, Tóth János: Differenciálegyenletek, 2004

[3] Macsotai Ágnes: Közönséges Differenciálegyenletek Kezdetiérték Feladatainak Numerikus Megoldása Matlab Alkalmazásával, 2010

[4] Frölich Georgina: A befogás stabilitása a korlátozott háromtest-problémában, 2004.

[5] Alexander L Godunov: Nonlinear Differential Equations and The Beauty of Chaos `http://ww2.odu.edu/~agodunov/teaching/notes/Cp05_chaos.pdf`

[6] David Houcque: Applications of MATLAB: Ordinary Differential Equations (ODE)

[7] `https://www.math.hmc.edu/~depillis/`

[8] Faragó István, Fekete Imre, Horváth Róbert: Numerikus módszerek példatár, 2013