Eötvös Loránd Tudományegyetem

Természettudományi Kar

Nagy Levente

Runge–Kutta módszerek implementálása és alkalmazása

BSc Elemző Matematikus Szakdolgozat

Témavezető:

Fekete Imre
Alkalmazott Analízis és Számításmatematikai Tanszék

Budapest
2019

Köszönetnyilvánítás

Szeretném kifejezni őszinte hálámat elsősorban témavezetőm, Fekete Imre felé, aki nemcsak hogy idejét és a hosszú évek munkája során megszerzett tapasztalatait osztotta meg velem, de emberileg és szakmailag is rendkívül inspirált.

Hatalmas köszönettel tartozom szüleimnek, testvéreimnek és a barátaimnak, akik hátteret biztosítottak munkámnak, és akik támogatása és szeretete nélkül sosem jutottam volna el idáig.

Végezetül pedig szeretném megköszönni gimnáziumi és egyetemi tanáraimnak, hogy megosztották velem tudásukat.

Tartalomjegyzék

1. fejezet

Bevezetés

„Minél alapvetőbb és minél nehezebben érthető egy új igazság, annál hatalmasabbak és jelentősebbek lesznek gyakorlati lehetőségei.”

Szent-Györgyi Albert

A differenciálegyenletek a természetben lezajló folyamatok és a mindennapokban tapasztalt folytonos változások – legyenek azok fizikai, kémiai, műszaki, közgazdasági stb. – leírásainak elengedhetetlen matematikai eszközei, ezért okkal született meg az igény arra, hogy komolyan foglalkozzunk azok megoldásaival. Az idők során rá kellett jönnünk, hogy csak speciális alakú differenciálegyenleteket tudunk pontosan megoldani, így olyan módszerek megalkotásaira szorulunk, mely eljárások során a végeredményhez nagyon hasonló, közelítő megoldásokat kapunk. Ezeket a numerikus számításokat a számítógépek elterjedése előtt papíron végezték, a XX. század közepétől viszont a számítógépeké lett a főszerep.

E dolgozat célja azon túlmenően, hogy egy átfogó képet adjon a differenciálegyenletek numerikus megoldási módszereinél felmerülő nehézségekről és azok áthidalásáról az, hogy a különböző módszerek gyakorlati alkalmazásait példákon és ábrákon keresztül is szemléltesse. A dolgozat felépítése az eltérő numerikus módszerek fejlődéseinek valós kronológiai hátterét próbálja tükrözni.

Nehéz lenne a differenciálegyenletekről és azok közelítő megoldási módszereiről értekezni, ha nem ismertetnénk a hozzájuk kapcsolódó fogalmakat és a kezdeti eredményeket, ezért a 2. fejezet ezen alapkövek részletezését helyezi előtérbe. Megismerkedünk a differenciál-

egyenletek közelítő megoldására szolgáló explicit Euler módszerrel, illetve annak javítási ötletével. Ennek a fejezetnek a [14] jegyzet szolgál alapjául. Mivel fő törekvésünk, hogy egyre pontosabb megoldásokat biztosító módszereket konstruáljunk, ezért a 3. fejezet a különböző eljárások pontossági rendjeinek növelését akadályozó tényezők kijavításáról, nagy matematikusok új ötleteinek ismertetéseiről és azok alkalmazásairól szól. Felvezetjük dolgozatunk központi szereplőjét, a Runge–Kutta módszercsaládot, melyet explicit (ERK) esetek után kiterjesztünk implicitre (IRK) is. A jól működő módszerek után azok pontosságának becslésével foglalkozunk, melynek nyomán eljutunk egy változó lépésközzel működő metódushoz is.

2. fejezet

Első- és másodrendű explicit egylépéses módszerek

Ahogy a bevezetőben is említettük, a dolgozatban numerikus módszereket szeretnénk implementálni, ezért ebben a fejezetben a differenciálegyenletek fogalmát és a közönséges differenciálegyenletek közelítő megoldásainak különböző, kezdetleges módszereit tárgyaljuk.

2.0.1. Definíció. *Az olyan egyenletet, amely ismeretlen függvények, annak független változói és az egyes változók szerinti (első vagy magasabb rendű) deriváltjai közötti kapcsolatot írja le, differenciálegyenletnek nevezzük.*

2.1. Közönséges differenciálegyenletek kezdetiérték-feladata

Azokat a differenciálegyenleteket, amelyek egy ismeretlen egyváltozós függvény és annak deriváltjai, valamint ugyanazon változóhoz tartozó ismert függvényei közötti kapcsolatot írnak le, közönséges differenciálegyenleteknek nevezzük. A továbbiakban az alábbi definíciókban használt jelöléseket fogjuk használni.

2.1.1. Definíció. *Legyen $G \subset \mathbb{R}^{d+1}$ összefüggő halmaz, f egy folytonos függvény. Közönséges differenciálegyenletnek mondjuk az $u'(t) = f(t, u(t))$ kifejezést.*

2.1.2. Definíció. *Legyen $G \subset \mathbb{R}^{d+1}$ egy tartomány (azaz összefüggő, nyílt halmaz), $(t_0, u_0) \in G$ egy adott pont ($t_0 \in \mathbb{R}, u_0 \in \mathbb{R}^d$), $f : G \to \mathbb{R}^d$ egy folytonos leképezés. Az*

$$u'(t) = f(t, u(t)) \tag{2.1}$$

$$u(t_0) = u_0 \tag{2.2}$$

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

Mivel – ahogy azt már korábban is említettük – a differenciálegyenletek a környezetünkben lezajló folyamatok leírásainak matematikai eszközei, ezért alapvető elvárás, hogy a felírt feladatnak létezzen egyértelmű megoldása. Egy Cauchy-feladat megoldása azt jelenti, hogy meghatározzuk az összes olyan $u : \mathbb{R} \to \mathbb{R}^d$ függvényt, amely valamely $I \subset \mathbb{R}$ intervallum pontjaiban behelyettesíthető a (2.1)-(2.2) feladatba, és ki is elégíti azt.

2.1.3. Definíció. *Az olyan $u : I \to \mathbb{R}^d$ (I egy nyílt intervallum) folytonosan differenciálható függvényt, amelyre*

- $\{(t, u(t)) : t \in I\} \subset G$;
- $u'(t) = f(t, u(t))$, *minden* $t \in I$;
- t_0 *és* $u(t_0) = u_0$

a (2.1)-(2.2) feladat megoldásának nevezzük.

A következő – bizonyítás nélkül közölt – tétel és a tételben felhasznált definíció biztosítja számunkra azt, hogy a kezdetiérték-feladatnak létezzen egyértelmű megoldása.

2.1.4. Definíció. *Legyen $G \subset \mathbb{R} \times \mathbb{R}^d$ tartomány. Az $f : G \to \mathbb{R}^d$ függvényt a második változójában Lipschitz-tulajdonságúnak nevezzük, ha létezik $L > 0$ úgy, hogy minden (t, p_1), $(t, p_2) \in G$ esetén*

$$|f(t, p_1) - f(t, p_2)| \leq L|p_1 - p_2|.$$

2.1.5. Tétel. (Picard-Lindelöf) *[13] Legyen $f : G \to \mathbb{R}^d$ folytonos függvény, ahol*

$$G = \{(t, u) \in \mathbb{R} \times \mathbb{R}^d : |t - t_0| \leq a \text{ és } |u - u_0| \leq b\}$$

henger, $(t_0, u_0) \in \mathbb{R} \times \mathbb{R}^d$ és $0 < a < \infty$, $0 < b < \infty$. Legyen $M = \max\limits_{(t,u)\in G} |f(t, u)|$, továbbá tegyük fel, hogy az f függvény második változójában Lipschitz-tulajdonságú. Ekkor a (2.1)-(2.2) kezdetiérték-problémának egyértelműen létezik megoldása a $[t_0 - \delta, t_0 + \delta]$ intervallumon, ahol $\delta = \min\{a, \frac{b}{M}\}$.

A továbbiakban feltesszük, hogy a tételben említett feltételek mindig teljesülnek.

2.1.6. Példa. [1] Tekintsük a fejlődő országok gazdasági növekedésének egy modelljét:

$$X(t) = \sigma K(t) \tag{2.3}$$

$$K'(t) = \alpha X(t) + H(t) \tag{2.4}$$

$$N(t) = N_0 e^{\rho t}, \tag{2.5}$$

ahol $X(t)$ a teljes évenkénti termelés, $K(t)$ a tőke mennyisége, $H(t)$ az évenként beáramló külföldi segítség, $N(t)$ pedig a lakosság száma a t időpillanatban. A (2.3) egyenletben szereplő σ skalár a tőke átlagos termelékenysége. A (2.4) egyenletben feltettük, hogy a tőke teljes növekedése a belső megtakarítás és a külföldi segélyek összege. Feltételezzük, hogy a megtakarítás arányos a termeléssel, ahol az α arányossági tényezőt *megtakarítási rátának* nevezzük. A (2.5) egyenlet pedig azt mondja ki, hogy a lakosság a ρ együtthatóval exponenciálisan nő.

Állítsunk fel differenciálegyenletet a tőke $K(t)$ mennyiségére! Tegyük fel, hogy $H(t) = H_0 e^{\mu t}$, $\alpha\rho \neq \mu$, és a $K(0) = K_0$ kezdeti érték mellett oldjuk meg a differenciálegyenletet! Határozzuk meg az egy főre jutó termelés $x(t) = \frac{X(t)}{N(t)}$ nagyságát!

Megoldás: A (2.3) és (2.4) alapján a $K(t)$ kielégíti a

$$K'(t) = \alpha\sigma K(t) + H(t) \tag{2.6}$$

lineáris differenciálegyenletet. Ha $H(t) = H_0 e^{\mu t}$, akkkor

$$K(t) = Ce^{\alpha\sigma t} + e^{\alpha\sigma t}\int e^{-\alpha\sigma t} H_0 e^{\mu t} dt = Ce^{\alpha\sigma t} + e^{\alpha\sigma t}\int H_0 e^{(\mu-\alpha\sigma)t} dt$$

$$= Ce^{\alpha\sigma t} + e^{\alpha\sigma t}\frac{H_0}{\mu-\alpha\sigma}e^{(\mu-\alpha\sigma)t} = Ce^{\alpha\sigma t} + \frac{H_0}{\mu-\alpha\sigma}e^{\mu}t.$$

Ha $t = 0$, akkor a $K(0) = K_0 = C + \frac{H_0}{\mu-\alpha\sigma}$, így $C = K_0 - \frac{H_0}{\mu-\alpha\sigma}$. A megoldás tehát

$$K(t) = \left(K_0 - \frac{H_0}{\mu-\alpha\sigma}\right)e^{\alpha\sigma t} + \frac{H_0}{\mu-\alpha\sigma}e^{\mu t}. \tag{2.7}$$

Az egy főre jutó termelés így $x(t) = \frac{X(t)}{N(t)} = \frac{\sigma K(t)}{N_0 e^{\rho t}}$. Ha felhasználjuk a (2.7)-ben szereplő $K(t)$-re vonatkozó összefüggést, akkor egyszerű számolás alapján

$$x(t) = x(0)e^{(\alpha\sigma-\rho)t} + \left(\frac{\sigma}{\alpha\sigma-\mu}\right)\frac{H_0}{N_0}e^{(\alpha\sigma-\rho)t}\left[1 - e^{(\mu-\alpha\sigma)t}\right].$$

$\diamondsuit$

2.2. Explicit egylépéses módszerek

Az esetek túlnyomó részében a közönséges differenciálegyenletek kezdetiérték-feladatainak megoldásai csak nagyon speciális f függvények esetén adhatók meg képletek segítségével. Az ilyesfajta megoldási törekvések helyett numerikus megoldást szeretnénk előállítani, ami azt jelenti, hogy az értelmezési tartománynak egyes pontjaiban az ismeretlen megoldásfüggvény értékeit véges számú lépéssel közelítőleg határozzuk meg. A továbbiakban kronológiailag haladva a különböző approximáló módszereket igyekszünk bemutatni. Olyan eljárásokat fogalmazunk meg, amelyben egy rögzített időpontban kiszámított közelítést egy korábbi időpontbeli közelítés felhasználásával határozunk meg. Ezeket a módszereket egylépéses módszereknek nevezzük. A célunk tehát a továbbiakban az

$$u'(t) = f(t, u(t)), \quad t \in [0, T]$$

$$u(0) = u_0$$

feladat egylépéses módszerekkel történő közelítő megoldása, ahol $T > 0$ olyan szám, amely mellett a (2.1)-(2.2) feladatnak létezik egyértelmű, megfelelően sima megoldása a $[0, T]$ intervallumon.

Explicitnek nevezzük azokat a módszereket, mely során a t_n pontbeli értéket ismerve a t_{n+1} pontbeli közelítés közvetlenül kiszámítható egy egyszerű helyettesítéssel. Kezdetben ilyen típusú módszereket ismertetünk.

2.2.1. Taylor-sorba fejtéses módszer

Ez az egyik legrégebben (XVIII. sz. elején) megkonstruált módszer. Tegyük fel, hogy az f függvény analitikus, vagyis léteznek tetszőleges rendű parciális deriváltjai. Ekkor az $u(t)$ megoldásfüggvény is analitikus, így akárhányszor differenciálható. A $t^* \in [0, T]$ pontban rendre a deriváltak a következők:

$$\begin{aligned} u'(t) &= f(t^*, u(t^*)), \\ u''(t) &= \partial_1 f(t^*, u(t^*)) + \partial_2 f(t^*, u(t^*))u'(t^*), \\ u'''(t) &= \partial_{11} f(t^*, u(t^*)) + 2\partial_{12} f(t^*, u(t^*))u'(t^*) + \\ &+ \partial_{22} f(t^*, u(t^*))u'(t^*)^2 + \partial_2 f(t^*, u(t^*))u''(t^*). \end{aligned} \tag{2.8}$$

Tegyük fel, hogy $t > t^*$ olyanok, amelyre $t \in [0, T]$ és $t^* \in [0, T]$. Mivel az $u(t)$ megoldásfüggvény analitikus, ezért Taylor-sora előállítja a t^* pont valamely környezetében. Tehát a

$$T_{n,u}(t) = \sum_{k=0}^{n} \frac{u^{(k)}(t^*)}{k!}(t - t^*)^k$$

Taylor-polinom $n \to \infty$ esetén konvergál az $u(t)$ megoldáshoz, ha t megfelelően közel van a t^* ponthoz. Ezért a konvergencia-tartományon belül a megoldás előállítható az

$$u(t) = \sum_{k=0}^{\infty} \frac{u^{(k)}(t^*)}{k!}(t - t^*)^k \tag{2.9}$$

öszefüggéssel.

A fejezet elején elvárásként fogalmaztuk meg, hogy a megoldást véges számú lépéssel közelítsük. Hogy ez megvalósuljon, ismertetjük az ún. lokális Taylor-módszert. Legyen h adott szám és generáljunk egy

$$\omega_h := \{0 = t_0 < h < 2h < \cdots < Nh = T\}$$

ún. ekvidisztáns (azonos lépésközű) rácshálót a $[0, T]$ intervallumon. A rácsháló pontjait, lépésközeit és finomságát a következőképpen jelöljük:

$$t_n = nh, \quad n = 0, 1, \ldots, N$$

$$h_n = t_{n+1} - t_n, \quad n = 0, 1, \ldots, N - 1$$

$$h = \max_n h_n$$

Mostantól tehát ezekben a t_n pontokban fogunk approximálni, ahol $u(t_n)$ közelítését y_n-nel, míg $u^{(k)}(t_n)$ közelítését $y_n^{(k)}$-val jelöljük, ahol $k = 0, 1, \ldots, p$. Az $y_0^{(k)}$ értékei a (2.8) összefüggések szerint pontosan kiszámíthatók a $t^* = 0$ behelyettesítéssel. Az $u(t_1)$ közelítését a következő képlettel tehát könnyedén kiszámíthatjuk:

$$u(t_1) \approx y_1 = \sum_{k=0}^{p} \frac{y_0^{(k)}}{k!} h_0^{(k)}.$$

Az $n = 1, 2, \cdots, N-1$ értékekre y_n ismeretében és a (2.8) összefüggések alapján közelítőleg meghatározhatjuk $y_n^{(k)}$ értékeit a $k = 0, 1, \cdots, p$ értékekre. Az $u(t_{n+1})$ közelítését tehát az alábbi képlettel határozhatjuk meg:

$$u(t_{n+1}) \approx y_{n+1} = \sum_{k=0}^{p} \frac{y_n^{(k)}}{k!} h_n^{(k)}.$$

A 2.1. táblázatban $p = 1, 2$ értékekre a módszerek a következőképpen néznek ki, midőn $n = 0, 1, \cdots, N-1$, és $y_0 = u_0$ adott:

p=1	$y_{n+1} = y_n + y_n' h_n = y_n + h_n f(t_n, y_n)$
p=2	$y_{n+1} = y_n + h_n y_n' + \frac{h_n^2}{2} y_n'' = y_n + h_n f(t_n, y_n) + \frac{h_n^2}{2}\big(\partial_1 f(t_n, y_n) + \partial_2 f(t_n, y_n) f(t_n, y_n)\big)$

2.1. táblázat. Módszerek $p = 1, 2$-re

Mindkét módszer esetén a p-edfokú Taylor-polinomot használjuk, ami megköveteli a parciális deriváltak kiszámítását. Ez sajnos már viszonylag kis p értékekre is nagyon sok munkát követel meg, így ez a módszer igencsak költséges úton jut el a közelítő megoldásig.

2.2.2. Explicit Euler módszer

A következő módszert Leonhard Euler svájci matematikus dolgozta ki és publikálta az *„Institutionum calculi integralis”* c. könyvében 1768-ban [2]. A módszer – ahogy azt a későbbiekben látni fogjuk – elsőrendű közelítést ad a kezdetiérték-feladat megoldására. Tekintsük a skaláris kezdetiérték-feladatot:

$$u'(t) = f(t, u(t)) \tag{2.10}$$

$$u(t_0) = u_0, \tag{2.11}$$

ahol $t \in (0, T)$. Legyen $t^* \in (0, T)$ rögzített, ω_h pedig egy ekvidisztáns rácsháló. Ekkor (2.10) és $t = t_n$ egyenlőség miatt

$$u'(t_n) = f(t_n, u(t_n)), \tag{2.12}$$

ezért ha $y_n \approx u(t_n)$, akkor a véges differenciás approximáció alapján (2.12) bal oldalára

$$u'(t_n) \approx \frac{y_{n+1} - y_n}{h} \tag{2.13}$$

teljesül, jobb oldalára

$$f(t_n, u(t_n)) \approx f(t_n, y_n). \tag{2.14}$$

Így tehát (2.13)-(2.14) alapján

$$\frac{y_{n+1} - y_n}{h} = f(t_n, y_n), \qquad \text{ahol } n = 0, 1, \ldots$$

Mivel y_0 adott ($y_0 \equiv u_0$), ezért

$$y_{n+1} = y_n + hf(t_n, y_n), \qquad \text{ahol } n = 0, 1, \ldots \tag{2.15}$$

Mivel y_{n+1} közvetlen kiszámítható y_n-ből, ezért ezt a módszert explicit Euler módszernek nevezzük. A lépések (2.15) alapján a következők:

$$\begin{aligned} y_0 &= u_0 \\ y_1 &= y_0 + hf(t_0, y_0) \equiv y_0 + hf(0, u_0) \\ y_2 &= y_1 + hf(t_1, y_1) \equiv y_1 + hf(h, y_1) \\ &\vdots \end{aligned}$$

MATLAB® program segítségével vizsgáljuk meg egy konkrét példán keresztül az explicit Euler módszer hatékonyságát. A módszerre megírt MATLAB® kódunk:

```
function [h,t,y]=expliciteuler(a,b,y0,N)

h=(b-a)/N;                  % lépésköz
t=linspace(a,b,N+1);        % az intervallum felosztása
y=zeros(1,N+1);             % numerikus megoldás vektora

%% Az explicit Euler módszer algoritmusa
y(1)=y0;
for j=1:N
        y(j+1)=y(j)+h*f(t(j), y(j));
end

%% Az f, vagyis az y'(t)=f(t,y(t)) egyenlet jobb oldala
function diffegy=f(t,y)

  diffegy=y+2*t+3;
```

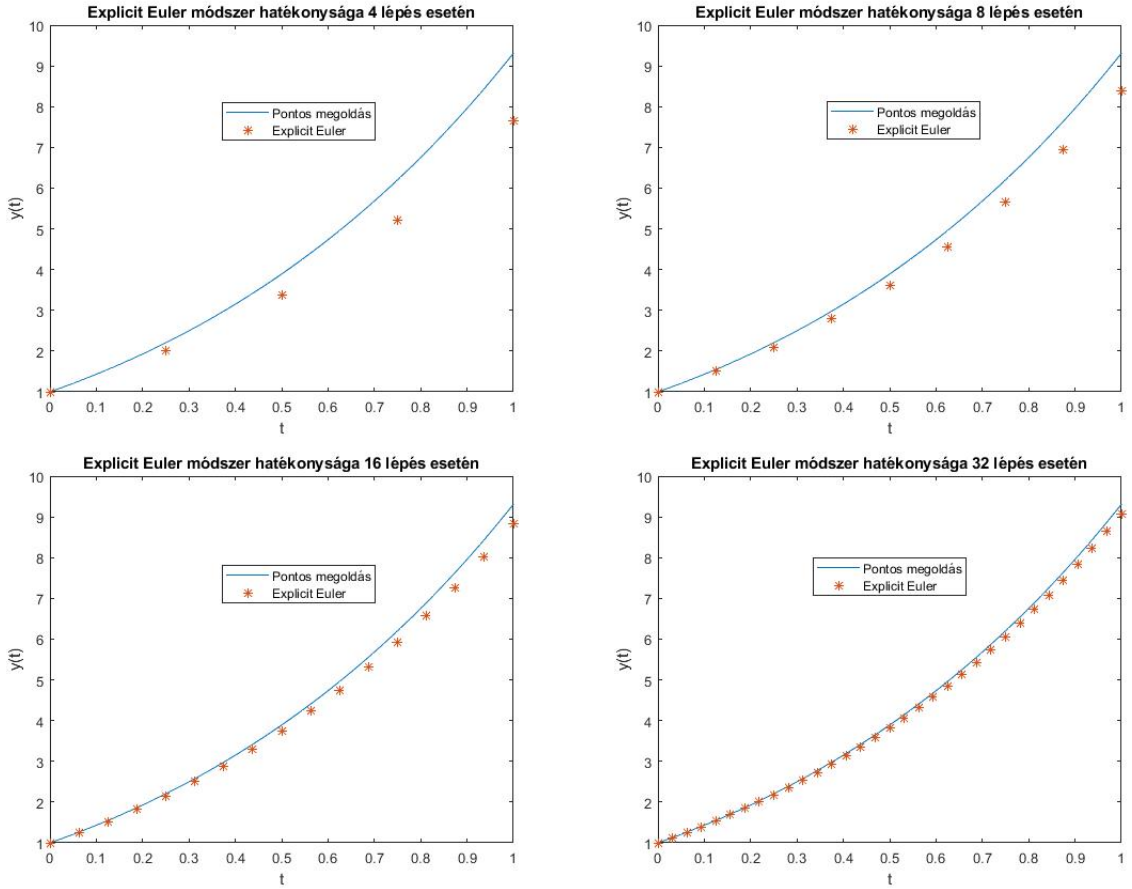
2.2.1. Példa. Tekintsük az

$$u'(t) = u(t) + 2t + 3, \quad t \in [0, 1]$$

$$u(0) = 1$$

feladatot, amelynek pontos megoldása $u(t) = 6e^t - 2t - 5$. ♢

A 2.1. ábrán látható, hogy az egyre finomodó rácshálón a módszer is egyre jobban közelíti a feladat pontos megoldását.



2.1. ábra. Az explicit Euler módszer finomodó lépésközű ekvidisztáns rácshálón

2.2.3. Alapfogalmak

Amikor numerikusan közelítő módszerek megkonstruálásáról beszélünk, egyúttal azt is vizsgálnunk kell, hogy a módszereink közelítő megoldása mekkora mértékben tér el az adott differenciálegyenlet pontos megoldásától. Az ebben az alfejezetben ismertetett definíciók ennek az eltérésnek a tanulmányozásában lesznek segítségünkre. Tekintsük továbbra is a ω_h ekvidisztáns rácshálót a $[0, T]$ intervallumon.

2.2.2. Definíció. *Azt mondjuk, hogy a numerikus módszer konvergens a $t^* \in \omega_h$ pontban, ha a következő összefüggés teljesül rá:*

$$\lim_{h \to 0} |y_n - u(t^*)| = 0. \tag{2.16}$$

Abban az esetben, ha (2.16) teljesül a $[0, T]$ intervallum minden t^* pontjában, akkor a numerikus módszert konvergensnek nevezzük az egész intervallumon.

2.2.3. Definíció. *A numerikus módszer p-ed rendben konvergens, ha*

$$|y_n - u(t^*)| = \mathcal{O}(h^p) \tag{2.17}$$

teljesül, ahol $p \geq 1$.

Vezessük be az $e_n := y_n - u(t^*)$ jelölést, ahol $u_n = u(t^*) \equiv u(t_n)$. Az e betű az angol „error" szóból származik, mely hibát jelent.

2.2.4. Definíció. *Az $e_n := |y_n - u(t_n)|$ rácsfüggvényt globális hibafüggvénynek nevezzük.*

A (2.16) összefüggés alapján a konvergencia feltétele:

$$\lim_{n \to \infty} e_n = 0.$$

Tekintsük az előző fejezetben ismertetett explicit Euler módszert:

$$\frac{y_{n+1} - y_n}{h} = f(t_n, y_n).$$

Alkalmazzuk (2.2.4) definíciót, majd rendezzük az egyenletet:

$$\frac{(e_{n+1} + u_{n+1}) - (e_n + u_n)}{h} = f(t_n, e_n + u_n)$$

Az egyenlőség jobb oldalához adjunk hozzá és vonjunk ki $f(t_n, u_n)$-t. Ekkor

$$\frac{e_{n+1} - e_n}{h} = -\frac{u_{n+1} - u_n}{h} + f(t_n, e_n + u_n).$$

$$\frac{e_{n+1} - e_n}{h} = \left[-\frac{u_{n+1} - u_n}{h} + f(t_n, u_n) \right] + \left[f(t_n, e_n + u_n) - f(t_n, u_n) \right].$$

A következő jelölések bevezetésével

$$\Psi_n^{(1)} := -\frac{u_{n+1} - u_n}{h} + f(t_n, u_n),$$
$$\Psi_n^{(2)} := f(t_n, e_n + u_n) - f(t_n, u_n)$$

az explicit Euler módszer hibaegyenlete felírható

$$\frac{e_{n+1} - e_n}{h} = \Psi_n^{(1)} + \Psi_n^{(2)}$$

alakban, ahol $\Psi_n^{(1)}$-t *lokális approximációs hibának* nevezzük. Ez azt mutatja meg nekünk, hogy a pontos megoldás milyen pontossággal elégíti ki a numerikus megoldást meghatározó egyenletet. A $\Psi_n^{(2)}$ tag pedig a pontos megoldás deriváltjának hibáját jellemzi az $u_n \approx y_n$ esetén.

2.2.5. Definíció. *Ha a numerikus módszer lokális approximációs hibájára teljesül, hogy*

$$\lim_{h \to 0} \Psi_n^{(1)} = 0,$$

akkor a módszert konzisztensnek nevezzük. A módszert p-ed rendben konzisztensnek hívjuk, ha $p > 0$-ra igaz, hogy

$$\Psi_n^{(1)} = \mathcal{O}(h^p).$$

Ezekkel az ismeretekkel felvértezve állapítsuk meg egy korábban megismert numerikus módszer, az explicit Euler módszer konzisztenciarendjét. A lokális approximációs hibafüggvény:

$$\Psi_n^{(1)} = -\frac{u_{n+1} - u_n}{h} + f(t_n, u_n) = -\frac{u(t_{n+1}) - u(t_n)}{h} + f(t_n, u(t_n)).$$

Fejtsük Taylor-sorba $u(t_{n+1})$-et t_n körül:

$$u(t_{n+1}) = u(t_n + h) = u(t_n) + hu'(t_n) + \mathcal{O}(h^2) \tag{2.18}$$

Továbbá, mivel

$$f(t_n, u(t_n)) = u'(t_n), \tag{2.19}$$

ezért (2.18)-t és (2.19)-t behelyettesítve kapjuk, hogy

$$\Psi_n^{(1)} = -\frac{(u(t_n) + hu'(t_n) + \mathcal{O}(h^2)) - u(t_n)}{h} + u'(t_n) = \mathcal{O}(h).$$

Az explicit Euler módszer tehát (megfelelően sima megoldás esetén) elsőrendben konzisztens.

2.2.6. Definíció. *Azt mondjuk, hogy a numerikus módszer konvergens a t^* pontban, ha*

$$\lim_{h\to 0} e_n(h) = 0.$$

Azt mondjuk, hogy p-ed rendben konvergens, amikor $e_n(h) = \mathcal{O}(h^p)$.

Az előző definíciókat felhasználva fontosnak tartjuk közölni a 2.2.7. Tételt, amely elégséges feltételt ad egy módszer konvergenciájára.

2.2.7. Tétel. *Ha egy egylépéses módszer p-ed rendben konzisztens és stabil (folytonosan függ a bemenő adatoktól), akkor a módszer konvergens.*

2.2.8. Megjegyzés. *A $\Psi_n^{(2)}$ tagra vonatkozó, második változójában teljesülő Lipschitzességi feltétel az explicit Euler módszer konvergencia bizonyítása során a stabilitáshoz szükséges.*

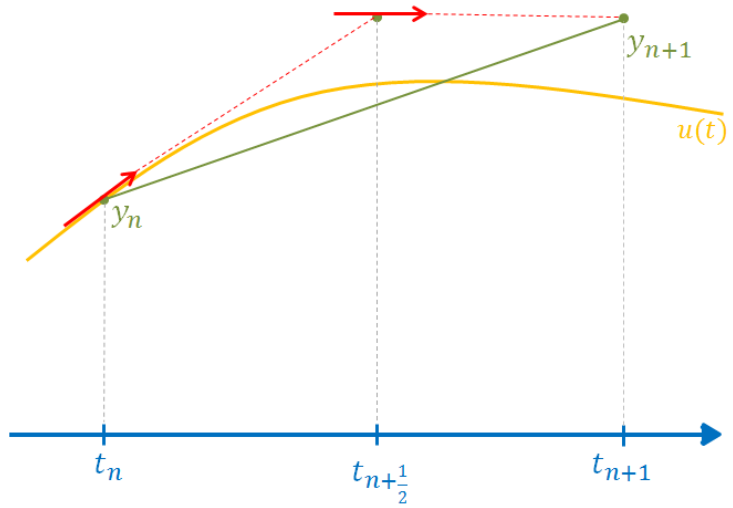
2.2.4. Kétlépcsős másodrendű explicit módszerek

Egynél magasabb rendű numerikus eljárás megkonstruálása a kezdetiérték-feladatra az említett egylépéses módszerek segítségével nehézségekbe ütközik. Bár a Taylor-féle módszerrel biztosíthatnánk magasabb rendű konzisztenciát, a kiszámítandó parciális deriváltak meghatározása meglehetősen bonyolult és költséges matematikai eljárást követel meg. A következőkben megmutatjuk, hogy ez a komplikált számításokat igénylő feladat egy frappáns ötlet segítségével kiküszöbölhető.

Tekintsük továbbra is az explicit Euler módszert:

$$y_{n+1} = y_n + hf(t_n, y_n), \quad \text{ahol } n = 0, 1, \ldots$$

Valamilyen módon javítani szeretnénk rajta, vagyis célunk az elsőrendű konzisztencia helyett a másodrendű konzisztencia biztosítása. Az ötlet az, hogy az eredeti képlettel csak egy fél, $h/2$ lépést tegyünk meg, majd a kapott $\left(t_{n+\frac{1}{2}}, y_{n+\frac{1}{2}}\right)$ pontban kiszámítjuk újra f értékét, és ezzel az értékkel tesszük meg az egész lépést (t_n, y_n) pontból a meghatározott irányban. Ezt szemlélteti a 2.2. ábra.



2.2. ábra. A javított Euler módszer alapgondolata

Érvényesek tehát a következő összefüggések:

$$y_{n+\frac{1}{2}} = y_n + \frac{1}{2}hf\left(t_n, y_n\right), \tag{2.20}$$

$$y_{n+1} = y_n + hf\left(t_{n+\frac{1}{2}}, y_{n+\frac{1}{2}}\right). \tag{2.21}$$

Ha (2.20)-t behelyettesítjük (2.21)-be, az alábbi egyenlőséghez jutunk:

$$y_{n+1} = y_n + hf\left(t_{n+\frac{1}{2}}, y_n + \frac{1}{2}hf\left(t_n, y_n\right)\right). \tag{2.22}$$

A lokális approximációs hibát vizsgálva választ kaphatunk arra a kérdésre, hogy javított-e az újonnan megkonstruált módszerünk az explicit Euler módszerhez képest. Ehhez tegyük fel, hogy f kétszer folytonosan differenciálható a [0,T] intervallumon, vagyis $f \in C^2([0,T])$.

$$\Psi_n^{(1)} = -\frac{u_{n+1} - u_n}{h} + f\left(t_{n+\frac{1}{2}}, u_n + \frac{1}{2}hf(t_n, u_n)\right) \tag{2.23}$$

Taylor-sorfejtéssel kapjuk, hogy

$$u_{n+1} = u(t_n + h) = u(t_n) + hu'(t_n) + \frac{h^2}{2}u''(t_n) + \mathcal{O}(h^3).$$

Az

$$f(x + \delta x, y + \delta y) = f(x, y) + \partial_1 f(x, y)\delta x + \partial_2 f(x, y)\delta y + \mathcal{O}((\delta x)^2, (\delta y)^2)$$

sorfejtést alkalmazva

$$f\left(t_{n+\frac{1}{2}}, u_n + \frac{1}{2}hf(t_n, u_n)\right) = f(t_n, u_n) + \partial_1 f(t_n, u_n)\frac{1}{2}h + \partial_2 f(t_n, u_n)\frac{1}{2}hf(t_n, u_n) + \mathcal{O}(h^2).$$

Mivel

$$u'(t) = f(t, u(t)) \text{ és}$$

$$u''(t) = \partial_1 f(t, u(t)) + \partial_2 f(t, u(t))u'(t),$$

az előbbieket (2.23)-be behelyettesítve kapjuk, hogy

$$\begin{aligned}\Psi_n^{(1)} &= -\left[u'(t_n) + \frac{h}{2}u''(t_n)\right] + O(h^2) + f(t_n, u_n) + \\ &+ \frac{h}{2}\left[\partial_1 f(t_n, u_n) + \partial_2 f(t_n, u_n) f(t_n, u_n)\right] + \mathcal{O}(h^2) = \\ &= -\left[u'(t_n) + \frac{h}{2}u''(t_n)\right] + u'(t_n) + \frac{h}{2}u''(t_n) + \mathcal{O}(h^2) = \mathcal{O}(h^2).\end{aligned}$$

Vagyis az új módszerünk másodrendű konzisztenciát biztosít. A (2.22)-t ezért javított Euler módszernek nevezzük. Vajon megadhatók egyéb másodrendű módszerek is? Tekintsük továbbra is a (2.10)-(2.11) kezdetiérték-feladatot. A $t = t^* + h$ pontban írjuk ki az $u(t)$ megoldás (2.9) alakú Taylor-sorának első pár tagját. Másodrendű konzisztenciát szeretnénk biztosítani, ezért a következő alakot írjuk fel:

$$u(t^* + h) = u(t^*) + hu'(t^*) + \frac{h^2}{2!}u''(t^*) + \mathcal{O}(h^3). \tag{2.24}$$

Felhasználva a (2.8) deriváltakat és bevezetve az $f \equiv f(t^*, u(t^*))$ jelölést (2.24) átírható az

$$\begin{aligned}u(t^* + h) &= u(t^*) + hf + \frac{h^2}{2!}(\partial_1 f + f\partial_2 f) + \mathcal{O}(h^3) \\ &= u(t^*) + \frac{h}{2}f + \frac{h}{2}[f + h\partial_1 f + hf\partial_2 f] + \mathcal{O}(h^3)\end{aligned} \tag{2.25}$$

alakra. Mivel

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

ezért (2.25) felírható a következőképpen:

$$u(t^* + h) = u(t^*) + \frac{h}{2}f + \frac{h}{2}f(t^* + h, u(t^*)) + \mathcal{O}(h^3). \tag{2.26}$$

További másodrendű módszerek kereséséhez általánosítsuk paraméteres alakban a (2.26) összefüggést:

$$u(t^*+h) = u(t^*)+b_1 hf(t^*, u(t^*))+b_2 hf(t^*+c_2 h, u(t^*)+a_{21}hf(t^*, u(t^*)))+\mathcal{O}(h^3), \tag{2.27}$$

ahol b_1, b_2, c_2 és a_{21} egyelőre tetszőleges paraméterek. Ha felírjuk a $t = t_n$ pontban a (2.27) egyenletet, akkor az alábbi numerikus módszert kapjuk:

$$y_{n+1} = y_n + b_1 h f(t_n, y_n) + b_2 h f(t_n + c_2 h, y_n + a_{21} h f(t_n, y_n)). \tag{2.28}$$

Ha Taylor-sorba fejtjük a (2.26) egyenlet jobb oldalát, akkor az

$$\begin{aligned} u(t^* + h) &= u(t^*) + b_1 h f + b_2 h \left[f + c_2 h \partial_1 f + a_{21} h f \partial_2 f \right] + \mathcal{O}(h^3) \\ &= u(t^*) + (b_1 + b_2) h f + h^2 \left[c_2 b_2 \partial_1 f + b_2 a_{21} f \partial_2 f \right] + \mathcal{O}(h^3) \end{aligned} \tag{2.29}$$

egyenlőséghez jutunk. Összevetve a (2.25) és (2.29) képleteket, azt látjuk, hogy a (2.28) által meghatározott numerikus módszer pontosan akkor másodrendű, ha

$$\begin{aligned} b_1 + b_2 &= 1 \\ c_2 b_2 &= 1/2 \\ a_{21} b_2 &= 1/2. \end{aligned} \tag{2.30}$$

Mivel ebben az egyenletrendszerben négy ismeretlenre van három egyenletünk, ezért a megoldás nem egyértelmű. Tetszőleges $b \neq 0$ esetén (2.30) megoldásai a következő alakban állnak elő:

$$\begin{aligned} b_2 &= b \\ b_1 &= 1 - b \\ c_2 = a_{21} &= \frac{1}{2b}. \end{aligned} \tag{2.31}$$

A 2.2 táblázatban láthatjuk, hogyan is fog kinézni (2.28), ha a b paraméter helyére konkrét értéket helyettesítünk be.

$b = 1$	$y_{n+1} = y_n + hf(t_n + \frac{h}{2}, y_n + \frac{h}{2} f(t_n, y_n))$
$b = \frac{1}{2}$	$y_{n+1} = y_n + \frac{1}{2} hf(t_n, y_n) + \frac{1}{2} hf(t_n + h, y_n + hf(t_n, y_n))$

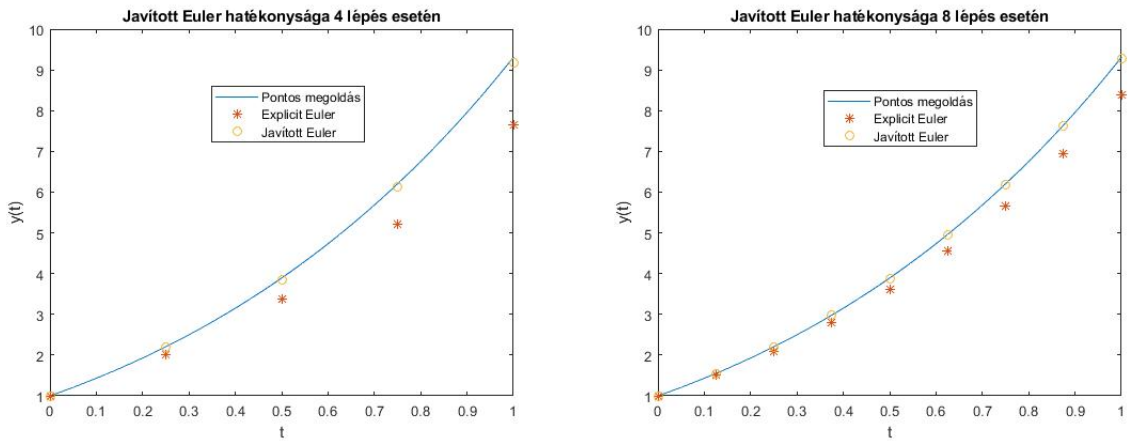
2.2. táblázat. Módszerek $b = 1,\ 1/2$ esetén

Vegyük észre, hogy $b = 1$ helyettesítésre a javított Euler módszert kaptuk vissza. A $b = \frac{1}{2}$-del nyert módszer a korábban tárgyaltak miatt szintén egy másodrendű közelítő módszer. Felmerülhet a kérdés: vajon lehetséges-e a b paraméter egy jobb megválasztásával nem csupán másodrendű, hanem harmadrendű konzisztenciát is biztosítani? A kérdésre a válasz nemleges, amellyel most részletesen nem fogunk foglalkozni.

Vizsgáljuk meg a 2.2.1. Példánkon keresztül, hogy a javított Euler módszer tényleg jobb közelítést biztosít-e. Ehhez a következő MATLAB® kódot írtuk meg:

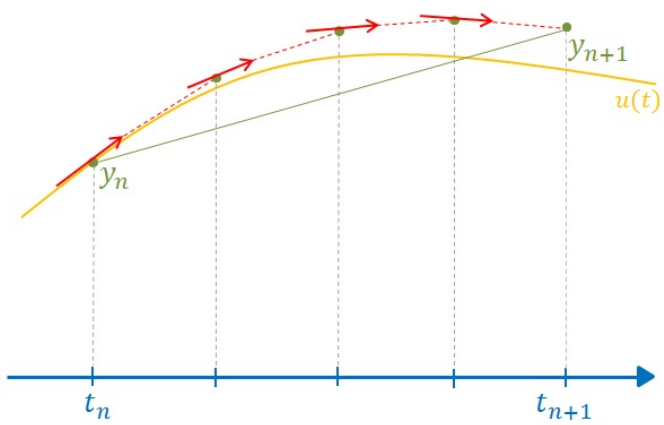
```
function [h,t,y]=javeuler(a,b,y0,N)

h=(b-a)/N;                  % lépésköz
t=linspace(a,b,N+1);        % az intervallum felosztása
y=zeros(1,N+1);             % numerikus megoldás vektora

%% Az javított Euler módszer algoritmusa
y(1)=y0;
for j=1:N
        y(j+1)=y(j)+h*f(t(j)+0.5*h, y(j)+0.5*h*f(t(j),y(j)));
end

%% Az f, vagyis az y'(t)=f(t,y(t)) egyenlet jobboldala
function diffegy=f(t,y)

 diffegy=y+2*t+3;
```

A 2.3. ábra alapján azt tapasztaljuk, hogy a javított Euler módszer valóban növeli az explicit Euler módszer hatékonyságát.



2.3. ábra. A javított Euler módszer finomodó lépésközű ekvidisztáns rácsháló

Amint azt az előzőekben láthattuk, ha egy $[t_n, t_{n+1}]$ nagyságú lépésen belül egy köztes pontban kiszámítjuk f értékét, majd a kezdőpontból a közbülső pont által meghatározott meredekséggel tesszük meg az egész lépést, akkor pontosabb közelítő megoldáshoz jutunk. Ahogy a 2.4. ábra szemlélteti, jogosan tehetjük fel a kérdést: vajon több ilyen belső lépcső felvételével magasabb konzisztenciarendet érhetünk el? A következő fejezetben egy általánosított módszerrel erre keressük a választ.



2.4. ábra. Növelhető-e több lépcsővel a konzisztenciarend?

3. fejezet

Runge–Kutta típusú módszerek

Ahogy már említettük, az ötlet az, hogy ékeljünk be további lépcsőket t_n és t_{n+1} közé. Karl Heun (1859-1929) 1900-ban [3] háromlépcsős, Martin Wilhelm Kutta (1867-1944) 1901-ben [4] pedig négylépcsős explicit módszert adott meg. Carl David Tolmé Runga (1856-1927) [5] és Kutta az 1900-as évek elején dolgoztak ki általános eljárást, melyet ma hagyománytiszteletből Runge–Kutta módszernek nevezünk.

3.1. Az s-lépcsős explicit Runge–Kutta módszer

A már említett lépcsőket k_i-k bevezetésével az alábbi rekurzív módon határozzuk meg:

$$\begin{aligned}
k_1 &= f(t_n, y_n) \\
k_2 &= f(t_n + c_2 h, y_n + h a_{21} k_1) \\
k_3 &= f(t_n + c_3 h, y_n + h(a_{31} k_1 + a_{32} k_2)) \\
&\vdots \\
k_s &= f(t_n + c_s h, y_n + h(a_{s1} k_1 + a_{s2} k_2 + \ldots + a_{ss-1} k_{s-1})),
\end{aligned} \tag{3.1}$$

ahol a_{ij}, c_i valós paraméterek. Az y_{n+1} értéket pedig ezen lépcsők lineáris kombinációjaként kaphatjuk meg:

$$y_{n+1} = y_n + h(b_1 k_1 + b_2 k_2 + \ldots b_s k_s), \qquad b_i \in \mathbb{R}.$$

3.1.1. Példa. Heun által konstruált harmadrendű módszer (ERK3):

$$\begin{aligned}
k_1 &= f\Big(t_n, y_n\Big) \\
k_2 &= f\left(t_n + \frac{1}{3}h, y_n + \frac{1}{3}hk_1\right) \\
k_3 &= f\left(t_n + \frac{2}{3}h, y_n + \frac{2}{3}hk_2\right) \\
y_{n+1} &= y_n + \frac{1}{4}hk_1 + \frac{3}{4}hk_3.
\end{aligned}$$

Kutta negyedrendű módszere (ERK4):

$$\begin{aligned}
k_1 &= f = f\Big(t_n, y_n\Big) \\
k_2 &= f\left(t_n + \frac{1}{2}h, y_n + \frac{1}{2}hk_1\right) \\
k_3 &= f\Big(t_n + h, y_n + hk_3\Big) \\
k_4 &= f\Big(t_n + h, y_n + hk_3\Big) \\
y_{n+1} &= y_n + \frac{1}{4}hk_1 + \frac{3}{4}hk_3.
\end{aligned}$$

$\diamondsuit$

Az a_{ij} paraméterekből egy $A \in \mathbb{R}^{s\times s}$ szigorú alsó háromszögmátrix, a c_i, b_i értékekből pedig $c, b \in \mathbb{R}^s$ vektorok írhatók fel:

$$\mathbf{A} = \begin{pmatrix} 0 & 0 & 0 & \dots & 0 \\ a_{21} & 0 & 0 & \dots & 0 \\ a_{31} & a_{32} & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ a_{s1} & a_{s2} & \dots & a_{ss-1} & 0 \end{pmatrix}, \qquad \mathbf{c} = \begin{pmatrix} c_1 \\ c_2 \\ c_3 \\ \vdots \\ c_s \end{pmatrix}, \qquad \mathbf{b} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_s \end{pmatrix}.$$

Vezessünk be egy új felírási módot, amely John C. Butcher (1933–) [6] új-zélandi matematikustól származik:

$$\begin{array}{c|ccccc}
0 & & & & & \\
c_2 & a_{21} & & & & \\
c_3 & a_{31} & a_{32} & & & \\
\vdots & \vdots & \vdots & \ddots & & \\
c_s & a_{s1} & a_{s2} & \dots & a_{ss-1} & \\
\hline
 & b_1 & b_2 & \dots & b_{s-1} & b_s
\end{array}$$

3.1.2. Definíció. *Egy explicit Runge–Kutta típusú módszer*

$$\begin{array}{c|c} \mathbf{c} & \mathbf{A} \\ \hline & \mathbf{b}^T \end{array}$$

alakban felírt paramétereinek táblázatát Butcher-tablónak nevezzük.

3.1.3. Példa. Megadjuk az eddig megismert négy numerikus módszerünk Butcher-tablóit, illetve Kutta ötödrendű eljárását, melyet végül Nyström (1925) [7] egészített ki.

Explicit Euler (ERK1)

$$\begin{array}{c|c} 0 & 1 \\ \hline & 1 \end{array}$$

Másodrendű kétlépcsős módszerek (ERK2)

$$\begin{array}{c|cc} 0 & 0 & 0 \\ 1/2b & 1/2b & 0 \\ \hline & (1-b) & b \end{array}$$

Heun (ERK3)

$$\begin{array}{c|ccc} 0 & 0 & 0 & 0 \\ 1/3 & 1/3 & 0 & 0 \\ 2/3 & 0 & 2/3 & 0 \\ \hline & 1/4 & 0 & 3/4 \end{array}$$

Kutta (ERK4)

$$\begin{array}{c|cccc} 0 & 0 & 0 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 & 0 \\ 1/2 & 0 & 1/2 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ \hline & 1/6 & 1/3 & 1/3 & 1/6 \end{array}$$

Kutta–Nyström (ERK5)

$$
\begin{array}{c|cccccc}
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1/3 & 1/3 & 0 & 0 & 0 & 0 & 0 \\
2/5 & 4/25 & 6/25 & 0 & 0 & 0 & 0 \\
1 & 1/4 & -3 & 15/4 & 0 & 0 & 0 \\
2/3 & 2/27 & 10/9 & -50/81 & 8/81 & 0 & 0 \\
4/5 & 2/25 & 12/25 & 2/15 & 8/75 & 0 & 0 \\
\hline
 & 23/192 & 0 & 125/192 & 0 & -27/64 & 125/192
\end{array}
$$

$\diamondsuit$

3.1.1. Rendfeltételek

Amikor a javított Euler módszeren túl egyéb másodrendű módszereket kerestünk, azt láttuk, hogy az adott c_2, a_{21}, b_1 és b_2 paraméterek (2.31) megválasztásával több másodrendű eljárást tudunk meghatározni. Ha $s = 3$ -ra felírjuk a (3.1) rekurziókat, akkor az alábbi – Butcher-tablóban összefoglalt – paraméterekhez jutunk:

$$
\begin{array}{c|ccc}
0 & & & \\
c_2 & a_{21} & & \\
c_3 & a_{31} & a_{32} & \\
\hline
 & b_1 & b_2 & b_3
\end{array}
$$

A módszer harmadrendűségéhez – a másodrendűséghez hasonlóan – a lokális approximációs hiba, $\Psi_n^{(1)}$ vizsgálata szükséges. Ebben az esetben azonban már sokkal többet kellene számolnunk, úgyhogy a módszer paramétereihez a szükséges feltételeket most levezetés nélkül közöljük:

$$
\begin{gathered}
c_2 = a_{21}, \quad c_3 = a_{31} + a_{32}, \\
c_3(c_3 - c_2) - a_{32}c_2(2 - 3a_2) = 0, \quad b_3 a_{32} c_2 = 1/6, \quad b_2 c_2 + b_3 c_3 = 1/2, \\
b_1 + b_2 + b_3 = 1.
\end{gathered}
\tag{3.2}
$$

Ez hat egyenletet jelent nyolc ismeretlenre. Ha jól megvizsgáljuk, láthatjuk, hogy Heun módszere (ERK3) megfelel a feltételeknek, vagyis tényleg harmadrendű közelítést biztosít.

3.1.4. Tétel. *Adott Butcher-tablójú explicit Runge–Kutta típusú módszer pontosan akkor konzisztens, amikor teljeseülnek a*

$$\boldsymbol{A}\boldsymbol{e} = \boldsymbol{c}, \quad \boldsymbol{b}^T\boldsymbol{e} = 1$$

feltételek, azaz

$$\sum_{k=1}^{s} a_{ik} = c_i, \quad i = 1, 2, \ldots, s \quad \text{és} \quad \sum_{k=1}^{s} b_k = 1.$$

A különböző módszerek rendfeltételeinek pontos meghatározását többen is megkísérelték az elmúlt évtizedek folyamán. A két legelterjettebb eljárás közül az egyik ún. Butcher-fákkal [8] dolgozik, melynek szellemi atyja a már említett John Butcher. A másik megközelítésben pedig Albrecht [9] és [10] cikkei lehetnek a segítségünkre. Mivel dolgozatunknak nem központi témája a konzisztenciarend szisztematikus származtatása, ezért ennek részletezésétől most eltekintünk.
A 3.1. táblázat foglalja össze azokat a rendfeltételeket, amelyek szükséges és elégséges feltételei az ötödrendűségnek. A táblázatban szereplő $C = \mathrm{diag}(c)$.

Rend (p)	Feltétel
1	$\mathrm{b}^T e = 1$
2	$\mathrm{b}^T c = 1/2$
3	$\mathrm{b}^T c^2 = 1/3,\ b^T Ac = 1/6$
4	$\mathrm{b}^T c^3 = 1/4,\ b^T CAc = 1/8,\ b^T Ac^2 = 1/12,\ b^T A^2 c = 1/24$
5	$\mathrm{b}^T c^4 = 1/5,\ b^T (c^T)^2 Ac = 1/10,\ b^T (AC)^2 = 1/20,\ b^T CAc^2 = 1/15,$ $\mathrm{b}^T Ac^3 = 1/20,\ b^T CA^2 c = 1/30, b^T ACAc = 1/40,$ $\mathrm{b}^T A^2 c^2 = 1/60,\ b^T A^3 c = 1/120$

3.1. táblázat. Ötödrendű feltételek

Láthatjuk, hogy a rend (p) növelésével a paraméterekre vonatkozó feltételek száma is igen gyorsan növekszik. A különböző rendekhez tartozó feltételek számát a 3.2. táblázat foglalja össze nekünk.

Rend (p)	1	2	3	4	5	6	7	8	9	10
Feltételek száma	1	2	4	8	17	37	85	200	486	1205

3.2. táblázat. Feltételek száma különböző p értékekre

A lenti MATLAB® program megadja a felhasználó által beírt módszer konzisztenciarendjét legfeljebb ötödrenddel bezárólag.

```
function rend(A,b)
c = sum(A,2); % c vektor
C = diag(c);  % C mátrix, diagonáljában a c vektor
%% Feltételek
p1 = [b'-1];
p2 = [b*c-1/2];
p3 = [b*c.^2-1/3, b*A*c-1/6];
p4 = [b*c.^3-1/4, b*C*A*c-1/8, b*A*c.^2-1/12, b*A^2*c-1/24];
p5 = [b*c.^4-1/5, (b.*c'.^2)*A*c-1/10, b*(A*c).^2-1/20, b*C*A*c
    .^2-1/15, b*A*c.^3-1/20, b*C*A^2*c-1/30, b*A*C*A*c-1/40, b*A
    ^2*c.^2-1/60, b*A^3*c-1/120];
%% Vizsgálat
if any(abs(p5)>eps)==0
    disp('Ötödrendű módszer')
elseif any(abs(p4)>eps)==0
    disp('Negyedrendű módszer')
elseif any(abs(p3)>eps)==0
    disp('Harmadrendű módszer')
elseif any(abs(p2)>eps)==0
    disp('Másodrendű módszer')
elseif any(abs(p1)>eps)==0
    disp('Elsőrendű módszer')
end
end
```

A 3.1.3. Példában látott módszerekre alkalmazva a fenti programot az elvárt konzisztenciarendeket kapjuk vissza (az ERK2 $b = 1$ esetén).

```
>> rend([1],[1])
Elsőrendű módszer
>> rend([0 0;1/2 0],[0 1])
Másodrendű módszer
>> rend([0 0 0;1/3 0 0;0 2/3 0],[1/4 0 3/4])
Harmadrendű módszer
>> rend([0 0 0 0;1/2 0 0 0;0 1/2 0 0;0 0 1 0],[1/6 1/3 1/3 1/6])
Negyedrendű módszer
>> rend([0 0 0 0 0 0;1/3 0 0 0 0 0;4/25 6/25 0 0 0 0;
1/4 -3 15/4 0 0 0;2/27 10/9 -50/81 8/81 0 0;
2/25 12/25 2/15 8/75 0 0],[23/192 0 125/192 0 -27/64 125/192])
Ötödrendű módszer
```

3.2. ERK módszerek alkalmazása rendszerekre

Elérkezett az idő, hogy az eddigi elméleti tapasztalatainkat most gyakorlati formába öntsük, és különböző példákon keresztül megmutassuk a megismert módszerek hatékonyságát. A 3.2.1. Példa egy olyan kétkomponensű differenciálegyenlet-rendszert tartalmaz, melynek a megoldását pontosan ismerjük. Ennek a pontos megoldásnak a segítségével tudjuk majd becsülni az alkalmazott módszer konvergenciarendjét.

3.2.1. Példa. (Keveredési probléma) Tekintsünk egy kétkomponensű oldódási/higítási folyamatot egy festék esetében. A tiszta anyag beáramlik adott sebességgel az első tartályba, majd az oldat ugyanakkora sebességgel áramlik tovább a második tartályba (a távozó festék mértéke arányos a koncentrációval). A folyadék térfogata mindkét tartályban konstans. Ekkor az egyes tartályokban a festék koncentrációjának változását leíró rendszer az alábbi:

$$K_1'(t) = -\frac{L}{V_1}K_1(t),$$
$$K_2'(t) = -\frac{L}{V_2}(K_2(t) - K_1(t)).$$

Legyenek

$$K_1(0) = 0,3, \quad K_2(0) = 0$$

$$L = 2, \quad V_1 = 10, \quad V_2 = 5,$$

ahol V_1 és V_2 az egyes tartályok térfogata, L pedig az arányossági tényező. ♢

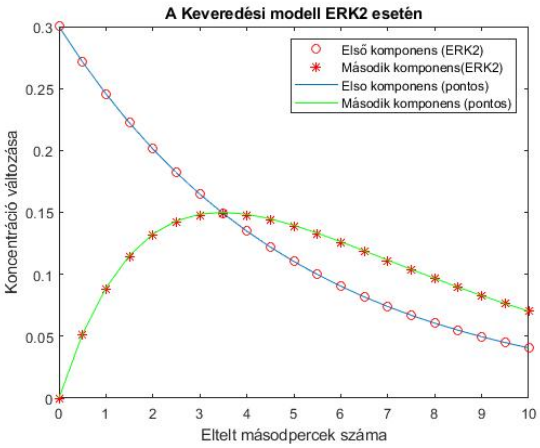
A feladat megoldásához terjesszük ki a már ismert javított Euler (ERK2) módszert rendszer esetre. A módszer MATLAB®-ban megírt kódja:

```
function [h,t,y]=ERK2_sys(a,b,y0,N)

h=(b-a)/N;                  % lépésköz
t=linspace(a,b,N+1);        % az intervallum felosztása
s=length(y0);               % a rendszer mérete
y=zeros(s,N+1);             % numerikus megoldás vektora

%% Az javított Euler módszer algoritmusa rendszerre
y(:,1)=y0;
for j=1:N
        y(:,j+1)=y(:,j)+h*f(t(j)+0.5*h, y(:,j)+0.5*h*f(t(j),y(:,
            j)));
end

function diffegy=f(t,y)
L=2; V1=10; V2=5;

diffegy=zeros(2,1);
diffegy(1)=-L/V1*y(1);
diffegy(2)=-L/V2*(y(2)-y(1));
```

A 3.2.1. Példa pontos megoldása

$$K_1(t) = 0,3 \cdot e^{-0,2t},$$
$$K_2(t) = 0,6 \cdot (e^{-0,2t} - e^{-0,4t}).$$

A 3.1. ábrán látható, hogy a numerikus módszerünk igen jó közelítését adja a feladat pontos megoldásának.



3.1. ábra. Az ERK2 módszer alkalmazása a festékkeveredési feladatra

3.2.1. Konvergenciarend becslése a pontos megoldásból

Tételezzük fel, hogy ismerjük a feladatunk pontos megoldását. Legyen ekkor e_h az ω_h rácshálón számított hiba. Jelölje tehát

$$e(h) = \|y(h) - \hat{y}(h)\|\,, \tag{3.3}$$

ahol $y(h)$ a numerikus megoldóvektor az ω_h rácson, $\hat{y}(h)$ pedig a pontos megoldás kiértékelése ugyanazon az ω_h rácson. Vezessük be a hiba diszkrét q-normáját az alábbi módon:

$$\|e\|_q = \left(h \sum_{i=1}^{N} |e_i| \right)^{1/q}.$$

Mivel $h^{1/q} \to 1$, ha $q \to \infty$, ezért a maximum-norma számítása

$$\|e\|_\infty = \max_{1 \le i \le N} |e_i|.$$

Ha a módszerünk p-ed rendű pontosságot biztosít, akkor azt várhatjuk, hogy

$$e(h) = Ch^p + \mathcal{O}(h^{p+1}).$$

Ha h kellően kicsi, akkor

$$e(h) \approx Ch^p. \tag{3.4}$$

A lépésköz felezésével

$$e(h/2) \approx C(h/2)^p$$

összefüggést kapjuk. Definiáljuk az

$$r_h = \frac{e(h)}{e(h/2)}$$

hibahányadost, így

$$r_h \approx 2^p,$$

vagyis

$$p \approx \log_2(r_h).$$

Általánosan, ha h_1 és h_2 különböző rácstávolságok, akkor azok alapján p értékét az alábbi módon becsülhetjük meg:

$$p \approx \frac{\log(e(h_1)/e(h_2))}{\log(h_1/h_2)}.$$

Tekintsük a 3.2.1. Példát. Mivel ismerjük a feladat pontos megoldását, ezért új ismereteinkkel felvértezve arra a kérdésre is választ tudunk adni, hogy mennyire „jó” a közelítés, amelyet az ERK2 biztosított. Az alábbi MATLAB® kód kiszámolja a hiba maximumnormáját:

```
%% Maximum norma
clear all
N=50;
for i=1:6
    [h,t,y]=ERK2_sys(0,10,[0.3 0],N*2^(i-1));
    pontos1=0.3*exp(-0.2*t);
    pontos2=0.6*(exp(-0.2*t)-exp(-0.4*t));
    norma1(i)=norm(pontos1-y(1,:),'inf');
    norma2(i)=norm(pontos2-y(2,:),'inf');
end
for i=1:length(norma1)-1
    rate1_max(i)=log2(norma1(i)/norma1(i+1));
    rate2_max(i)=log2(norma2(i)/norma2(i+1));
end
rate1_max
rate2_max
rate_max=min(rate1_max(end), rate2_max(end))
```

Futtatva a kódot a következő eredményt kapjuk:

```
rate1_max =
    2.0218    2.0109    2.0054    2.0027    2.0014
rate2_max =
    2.0491    2.0243    2.0121    2.0061    2.0030
rate_max =
    2.0014
```

Az első sor az első komponensre vonatkozó egymás utáni hibahányadosokból becsült konvergenciarend. A második sor hasonlóan az előzőhöz, viszont a második komponensre értendő, az utolsó sor pedig a rendszerre vonatkozó érték.

Az analóg módon megírt 1-es és 2-es normákkal kiegészítve az alábbi eredményt kapjuk:

```
rate1_1 =
    2.0281    2.0141    2.0070    2.0035    2.0018
rate2_1 =
    2.0545    2.0271    2.0135    2.0068    2.0034
rate_1 =
    2.0018
rate1_2 =
    2.0246    2.0122    2.0061    2.0031    2.0015
rate2_2 =
    2.0518    2.0256    2.0128    2.0064    2.0032
rate_2 =
    2.0015
```

Az eredményekből egyértelműen látszik, hogy az alkalmazott módszerünk másodrendű konvergenciát biztosít.

3.2.2. Példa. (SIR modell) Vegyük a járványterjedés legalapvetőbb, ún. SIR modelljét:

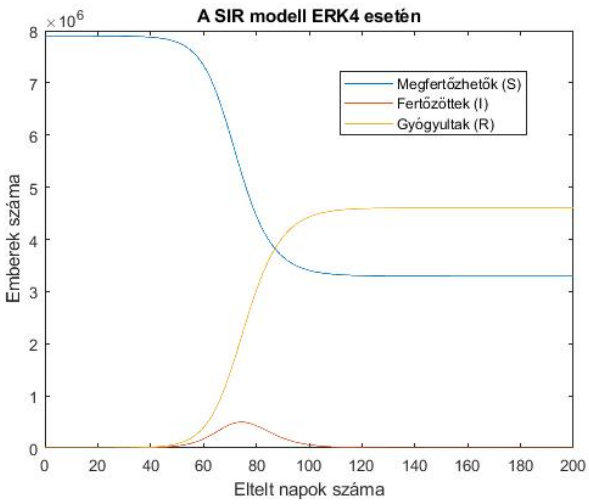
$$\begin{aligned} S'(t) &= -\alpha S(t)I(t), \\ I'(t) &= \alpha S(T) - I(t) - \beta I(t), \\ R'(t) &= \beta I(t), \end{aligned}$$

ahol S a fertőzhetők, I a fertőzöttek, R pedig a gyógyultak számát mutatja az idő függvényében. Az $\alpha > 0$ paraméter a fertőzési hányados, míg a $\beta > 0$ paraméter a fertőzők eltávolításának rátája. Tekintsük 1968 decemberének végén New York városát, melyet elért a hongkongi influenza pandémia. Tételezzük fel, hogy New York-ban kezdetben 7,9 millió ember volt egészséges, mindössze 10 ember volt fertőzött. A gyógyultak száma eleinte 0. A mért adatok alapján $\alpha = 1/2$ és $\beta = 1/3$. ♢

A megoldáshoz most a negyedrendű ERK4 módszert fogjuk használni, melynek kódja az ERK2 módszerétől az alábbi sorokban tér el:

```
%% A RK4 módszer algoritmusa

y(:,1)=y0; % kezdeti érték megadása
for j=1:N
        k1=f(t(j),y(:,j));
        k2=f(t(j)+0.5*h,y(:,j)+0.5*h*k1);
    k3=f(t(j)+0.5*h,y(:,j)+0.5*h*k2);
        k4=f(t(j)+h,y(:,j)+h*k3);
        y(:,j+1)=y(:,j)+h*(1/6*k1+1/3*k2+1/3*k3+1/6*k4);
end
```

A feladat megoldását szemléltető 3.2. ábráról azt olvashatjuk le, hogy kb. a 100-adik nap után tűnt el teljesen az influenza az emberek köréből.



3.2. ábra. Az ERK4 módszer alkalmazása a SIR modellre

A 3.2.2 Példában nem ismerjük a feladat pontos megoldását, így nem tudjuk alkalmazni az alfejezetben ismertetett módszert a konvergenciarend becslésére. Más eljárással számításainkat azonban el fogjuk tudni végezni.

3.2.2. Konvergenciarend becslése finommegoldásból

Tegyük fel, hogy nem ismerjük a pontos megoldást, viszont megtehetjük azt, hogy a módszert egy nagyon finom felosztású $\omega_{\bar{h}}$ rácshálón futtassuk. Az ω_h legyen olyan, hogy minden pontját tartalmazza az $\omega_{\bar{h}}$ rácsháló. Az összehasonlításhoz az $\omega_{\bar{h}}$ rácson kiszámított $y(\bar{h})$ megoldást kell leszűkíteni az ω_h durva rácsra. Az így leszűkített $\bar{y}(h)$ megoldást hasonlítjuk össze $y(h)$-val. Ekkor a (3.3) hibavektorra közvetlen számolással az

$$e(h) \leq \|y(h) - \bar{y}(h)\| + \|\hat{y}(h) - \bar{y}(h)\|$$

becslés adódik. Ha $\bar{h}$ jóval kisebb, mint h, akkor a fenti becslésben a második tag elhagyható. Ez a becslés azt jelenti, hogy a 3.2.1. fejezetben ismertetett technikát alkalmazhatjuk.

A 3.2.2. Példában először alkalmazzuk az ERK4 numerikus módszert kellően nagy, 2^{16} lépésszám mellett, majd a közelítő eredményt fogadjuk el pontos megoldásnak (finommegoldás). A konvergenciarend meghatározásához pedig számoljuk ki a finommegoldás és a nagyobb lépésközű közelítések közti eltérést. Az ellenőrző MATLAB® kód 1-es normában a következő:

```
%% Egyes norma
[h,t,y]=ERK4sys_SIR(0,200,[7900000 10 0],2^16);
finommo=y;
size(finommo);

N=2^6;
for i=1:5
    [h,t,y]=ERK4sys_SIR(0,200,[7900000 10 0],N*2^(i-1));
    pontos1=finommo(1,1:2^(11-i):2^16+1);
    pontos2=finommo(2,1:2^(11-i):2^16+1);
    pontos3=finommo(3,1:2^(11-i):2^16+1);
    lepeskoz(i)=h;
    norma1(i)=lepeskoz(i)*sum(abs(pontos1-y(1,:)));
```

```
    norma2(i)=lepeskoz(i)*sum(abs(pontos2-y(2,:)));
    norma3(i)=lepeskoz(i)*sum(abs(pontos3-y(3,:)));
end
for i=1:length(norma3)-1
    rate1_1(i)=log2(norma1(i)/norma1(i+1));
    rate2_1(i)=log2(norma2(i)/norma2(i+1));
    rate3_1(i)=log2(norma3(i)/norma3(i+1));
end
rate1_1
rate2_1
rate3_1
rate_1=min([rate1_1(end); rate2_1(end); rate3_1(end)])
```

Az előző példához hasonlóan értelmezhetjük az egyes komponensekre vonatkozó eredményeket.

```
rate1_1 =
    3.7155    3.8550    3.9269    3.9633
rate2_1 =
    3.7068    3.8542    3.9261    3.9631
rate3_1 =
    3.7168    3.8556    3.9272    3.9635
rate_1 =
    3.9631
```

Az analóg módon megírt 2-es és maximum-normával kiegészítve az eredmény az alábbi:

```
rate1_2 =
    3.7052    3.8514    3.9254    3.9626
rate2_2 =
    3.7056    3.8516    3.9255    3.9627
rate3_2 =
    3.7068    3.8521    3.9257    3.9628
rate_2 =
    3.9628
```

```
rate1_max =
    3.7012    3.8500    3.9248    3.9624
rate2_max =
    3.7058    3.8516    3.9256    3.9621
rate3_max =
    3.7067    3.8522    3.9256    3.9627
rate_max =
    3.9621
```

A finommegoldással történő eljárás sikeres stratégiának bizonyult, az eredményről pedig egyértelműen leolvasható, hogy az ERK4 módszer biztosítja a várt negyedrendű konvergenciát.

3.2.3. Megjegyzés. A 3.2.1. és 3.2.2. alfejezetek becslésre vonatkozó részei a LeVeque könyv [11] gondolatait veszik alapul.

3.3. Implicit Runge–Kutta módszerek

Egy Runge–Kutta típusú módszer Butcher-tablóját egy $\mathbf{A} \in \mathbb{R}^{s\times s}$ mátrix és $\mathbf{c}$,$\mathbf{b} \in \mathbb{R}^s$ vektorok elemeivel definiáltuk. Ha figyelmesen megnézzük az eddig ismertetett, explicit módszerek Butcher-tablóit, azt a megállapítást tehetjük, hogy $\mathbf{A}$ szigorú alsó háromszög mátrix. Az olyan módszereket, melyekre $\mathbf{A}$ nem szigorú alsó háromszögmátrix, *implicit Runge–Kutta (IRK) típusú módszereknek* nevezzük.

Az IRK módszerek esetén k_i értékének kiszámolása egy s ismeretlenes (általában nemlineáris) egyenletrendszer megoldását igényli. Ez az eljárás alkalmazását bonyolultabbá teszi, azonban ugyanazon lépésszám mellett az ERK módszerekhez képest magasabb konzisztencia és jobb stabilitási tulajdonságok biztosíthatók. Az IRK módszer Butcher-tablója a következő:

$$\begin{array}{c|ccccc}
c_1 & a_{11} & a_{12} & \dots & a_{1s-1} & a_{3s} \\
c_2 & a_{21} & a_{22} & \dots & a_{2s-1} & a_{3s} \\
c_3 & a_{31} & a_{32} & \dots & a_{3s-1} & a_{3s} \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
c_s & a_{s1} & a_{s2} & \dots & a_{s-1} & a_s \\
\hline
 & b_1 & b_2 & \dots & b_{s-1} & b_s
\end{array}$$

Az s-lépcsős implicit IRK módszer felírása pedig:

$$\begin{aligned}
k_1 &= f(t_n + c_1 h, y_n + h(a_{11}k_1 + a_{12}k_2 + \ldots + a_{1s}k_s)) \\
k_2 &= f(t_n + c_2 h, y_n + h(a_{21}k_1 + a_{22}k_2 + \ldots + a_{2s}k_s)) \\
&\vdots \\
k_s &= f(t_n + c_s h, y_n + h(a_{s1}k_1 + a_{s2}k_2 + \ldots + a_{ss}k_s)),
\end{aligned}$$

ahol a_{ij}, c_i valós paraméterek. Az y_{n+1} értéket pedig ezen lépcsők lineáris kombinácójaként kaphatjuk meg:

$$y_{n+1} = y_n + h(b_1k_1 + b_2k_2 + \ldots b_sk_s), \qquad b_i \in \mathbb{R}.$$

3.3.1. Példa. (Implicit Euler) Legyen a módszer Butcher-tablója

$$\begin{array}{c|c}
1 & 1 \\
\hline
 & 1
\end{array}$$

alakú. Ez tehát egy egylépcsős implicit Runge–Kutta típusú módszer, amely részletesen kiírva

$$\begin{aligned}
k_1 &= f(t_n + h, y_n + hk_1) \\
y_{n+1} &= y_n + hk_1.
\end{aligned}$$

Az első lépésben megoldjuk k_1 ismeretlenre az első egyenletet valamilyen iterációs módszer alkalmazásával, majd a megoldást behelyettesítjük a második képletbe. A módszer elsőrendű konzisztenciát biztosít. ♢

3.3.2. Példa. (Trapéz-módszer) A módszer Butcher tablója

$$\begin{array}{c|cc}
0 & 0 & 0 \\
1 & 1/2 & 1/2 \\
\hline
 & 1/2 & 1/2
\end{array}$$

alakú. Ez egy kétlépcsős implicit Runge–Kutta módszer, amely a következőt jelenti:

$$\begin{aligned}
k_1 &= f\Big(t_n, y_n\Big) \\
k_2 &= f\left(t_n + h, y_n + \frac{1}{2}k_1 + \frac{1}{2}k_2\right) \\
y_{n+1} &= y_n + \frac{1}{2}hk_1 + \frac{1}{2}hk_2.
\end{aligned}$$

A fenti módszer másodrendű. ♢

Az alapvető különbség az explicit és az implicit módszerek között, hogy míg explicit esetben y_n ismeretében közvetlenül ki tudjuk számolni y_{n+1} értékét, addig az implicit esetben minden időpillanatban egy nemlineáris egyenletet, vagy – rendszer esetén – egyenletrendszert kell megoldanunk. Ezek hatékony megoldására valamilyen iterációs módszert, többnyire a specifikusan módosított Newton-módszert szoktuk használni.

3.3.1. Merev feladatok

Az eddigi feladatok megoldásaiban nem volt tranziens szakasz, azaz olyan részidő-intervallum, amely eltérő skálájú lépésközt igényelt volna. Ugyanakkor gyakran adódnak olyan problémák, amelyek többrétegű skálázást *(multiscale)* követelnek. Például a megoldásfüggvény egy szakasza időben sokkalta gyorsabban zajlik le, mint a másik. Explicit módszerek esetén, stabilitási tulajdonságokat figyelembe véve ez viszont nagymértékű korlátozást jelent a h lépésközre nézve. Ekkor h olyan kicsi is lehet, hogy akár a gépi nulla alá is kerülhetünk. Ez az adott numerikus módszert alkalmazhatatlanná teszi. Az ilyen típusú feladatokat nevezzük *merev feladatoknak*. Precízebb definícióért és a fogalom 60 éves történelmi fejlődéséért lásd [16]. Merev differenciálegyenlet-rendszerekre az implicit módszerek jobb stabilitási tulajdonságai miatt nagyobb h lépésközt engedhetünk meg. Ugyanakkor valós feladatok esetén az ekvidisztáns rács nem kellően hatékony. Tekintsünk egy ismert merev problémát.

3.3.3. Példa. [15] Balthazar van der Pol (1889-1959) 1927-ben egy vákuumcsövet tartalmazó egyszerű áramkör vizsgálata közben időnként zajokat figyelt meg. Az áramkört modellező egyenlet valóban öngerjesztett rezgéseket mutatott bizonyos μ paramétertartományokban (ami meglepő volt, hiszen explicit időfüggést nem tartalmaz az egyenlet). A feladatot leíró rendszer:

$$\begin{aligned} u_1'(t) &= u_2(t) \\ u_2'(t) &= \mu(1 - u_1(t)^2)u_2(t) - u_1(t). \end{aligned}$$

Legyen $\mu = 100$ és $u(0) = [2\ 0]^T$, az egyenletet pedig tekintsük a $T = [0, 300]$ időintervallumban. $\diamondsuit$

Implicit módszerek esetén tudjuk, hogy minden lépésben egy nemlineáris egyenletrendszert kell megoldanunk. Gyakorlatban ezt a Newton-módszer egy variánsával valósítjuk

meg. A [18] alapján, ennek szemléltetésére tekintsük a Trapéz-módszert. Írjuk át a következő alakra:

$$g(y_{n+1}) = y_{n+1} - y_n - \frac{1}{2}h\Big(f(t_n, y_n) + f(t_{n+1}, y_{n+1})\Big) = 0.$$

Ekkor g Jacobi mátrixa a

$$J = \frac{\partial g}{\partial y_{n+1}} = I - h\tilde{J}\Big|_{(t_{n+1}, y_{n+1})},$$

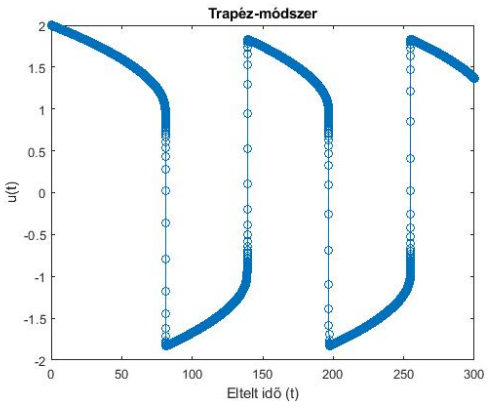
ahol $\tilde{J}$ a differenciálegyenlet jobb oldalához tartozó alkalmas Jacobi-mátrix.

A fenti eljárás alapján megírt Trapéz-módszer MATLAB® kódja, amellyel a 3.3.3. Példát szeretnénk megoldani az alábbi:

```
function [t,y,it]=trapezsys(a,b,y0,N,TOL,maxit)
%% Előkészületek
h=(b-a)/N;
t=linspace(a,b,N+1);
m=length(y0);
y=zeros(m,N+1);
y(:,1)=y0;
%% Trapéz módszer
for j=1:N
        yo=y(:,j);
        yj=yo;          % Kezdő Newton
        it=0;           % Newton lépésszámláló
        flag=1;         % Biztonsagi flag (1 oké, 0 nem oké)
        while flag
                yc=yj;
                [f,J]=diffegy (t(j+1),yc);
                [f1]=diffegy2 (t(j),yo);
                Jacobi=eye(m)-h*J;
                g=yc-yo-h*f;
                dy=Jacobi\g; % dy=y_{j+1}-y_j
                yj = yc - dy;
                it=it+1;
```

```
                    if norm(dy) < TOL * norm(yj) || it == maxit
                            flag=0;
                    end
            end
            y(:,j+1)=yj;
end
function [f,J]=diffegy(t,y)
%% Kimenő paraméter
% f        az egyenlet jobboldala
% J        az f Jacobi mátrixa
mu=100;
f = [y(2); mu*(1-y(1)^2)*y(2)-y(1)];
if nargout > 1
J = zeros(2,2);
J(1,1) = 0;                    J(1,2) = 1;
J(2,1) = -1-mu*2*y(1)*y(2);  J(2,2) = mu*(1-y(1)^2);
end
function [f1]=diffegy2(t,y)
mu=100;
f1 = [y(2); mu*(1-y(1)^2)*y(2)-y(1)];
```

A 3.3. ábrán láthatjuk, hogy az implicit Runge–Kutta típusú Trapéz-módszer alkalmazásával a merev van der Pol egyenlet megoldását is pontosan meg tudtuk közelíteni.



3.3. ábra. A Trapéz-módszer megoldása a van der Pol egyenletre

A merev feladatok sajátossága, hogy az eddigiekhez képest az elvárt konvergenciarend nem minden esetben teljesül, ún. *rendcsökkenés* következik be. Ennek részletezésével nem foglalkozunk, viszont megjegyezzük, hogy speciális Runge–Kutta módszerekkel a probléma kezelhető [17].

3.4. Változó lépésköz

Merev feladatok megoldásánál amellett, hogy az explicit módszerek rosszul működnek, rögzített h lépéstávolságok esetén az implicit eljárások sem mindig hatékonyak. A probléma feloldása a h lépésköz adaptív megválasztásában keresendő, amely során a módszer érzékelni tudja a merev feladatokra jellemző „kilengéseket”.

A változó lépésközök megválasztásának egy módja, hogy két különböző rendű numerikus módszer megoldásait hasonlítjuk össze. A két módszert célszerű úgy megválasztani, hogy rendjeik p és $p+1$ legyenek. Ezen módszerek megoldásait jelölje $y^p(h)$ és $y^{p+1}(h)$. Ekkor a lokális hibát a két módszer különbségéből becsüljük. Bővítsük ki az adott $(p+1)$-ed rendű Runge–Kutta típusú módszer Butcher-tablóját egy nála kisebb, p-ed rendet biztosító, b_i értékeket tartalmazó vektorral. A Butcher-tabló a következőképpen néz ki:

$$\begin{array}{c|c} \mathbf{c} & \mathbf{A} \\ \hline & \mathbf{b}^T \\ & \hat{\mathbf{b}}^T \end{array}$$

A $\mathbf{b}$ jelöli a p-ed rendű módszer, $\hat{\mathbf{b}}$ pedig a $(p+1)$-ed rendű módszer súlyait tartalmazó vektort. Ekkor azt mondjuk, hogy az alacsonyabb rendű módszer *beágyazott módszere* a magasabb rendű Runge–Kutta típusú módszernek. Az ilyen módszerpárok esetén a hibát (3.3) alapján

$$e(h_n) = \left\| h_n \sum_{i=1}^{s} (b_i - \hat{b}_i) k_i \right\|$$

módon tudjuk becsülni, ahol b_i és $\hat{b}_i$ a $\mathbf{b}$ és a $\hat{\mathbf{b}}$ vektorok megfelelő komponensei.

Az adott TOL pontosság eléréséhez az

$$\|\hat{y}(h_n) - y(h_n)\| \leq \mathrm{TOL} \tag{3.5}$$

egyenlőtlenséget kell ellenőriznünk. Ha (3.5) teljesül, akkor $y(h_n)$ értékét elfogadjuk az új közelítésnek, ha (3.5) nem teljesül, akkor egy másik, az eddiginél kisebb h_{n+1} lépéshosszt kell választani, amelyre $e(h_n) \approx Ch_{n+1}^{p+1} \leq \mathrm{TOL}$. Mivel $e(h_n) \approx Ch^{p+1}$, ezért

$$\frac{Ch_{n+1}^{p+1}}{Ch_n^{p+1}} \leq \frac{\mathrm{TOL}}{e(h_n)},$$

ahonnan a

$$h_{n+1} = \left(\frac{\mathrm{TOL}}{e(h_n)}\right)^{\frac{1}{p+1}} \cdot h_n$$

összefüggést kapjuk.

A magasabb rendű $\hat{\mathbf{b}}$ módszerrel lépünk tovább, amit a szakirodalomban [12] XEPS *(error-per-step, local extrapolation)* módszernek neveznek.

3.4.1. Példa. Tekintsük a következő merev differenciálegyenlet-rendszert:

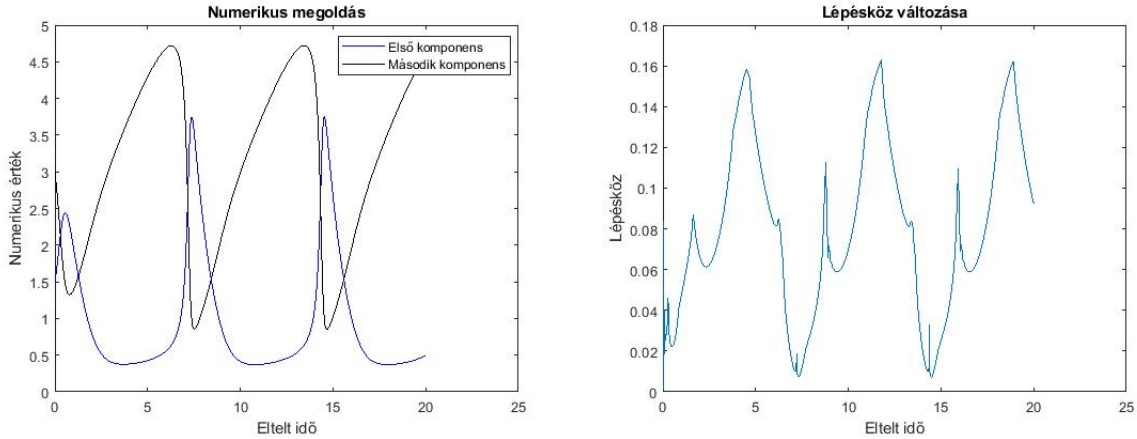
$$\begin{aligned} u_1'(t) &= 1 + u_1^2(t)u_2(t) - 4u_1(t) \\ u_2'(t) &= 3u_1(t) - u_1^2(t)u_2(t). \end{aligned}$$

Ez egy bizonyos autokatalikus reakció elméleti modellje, melyet röviden Brusselator modellnek hívunk. A fenti rendszerben szereplő konstansok alkalmas megválasztásai az általános modellben lévő paraméterekre nézve. ♢

A 3.4.1. Példa megoldásához írjuk fel a beágyazott módszerrel kiegészülő Butcher-tablót, ahol $\mathbf{b}$ egy harmadrendű, $\hat{\mathbf{b}}$ pedig egy negyedrendű ERK módszert jelöl.

$$\begin{array}{c|ccccc} 0 & 0 & 0 & 0 & 0 & 0 \\ 1/3 & 1/3 & 0 & 0 & 0 & 0 \\ 2/3 & -1/3 & 1 & 0 & 0 & 0 \\ 1 & 1 & -1 & 1 & 0 & 0 \\ 1 & 1/8 & 3/8 & 3/8 & 1/8 & 0 \\ \hline & 1/12 & 1/2 & 1/4 & 0 & 1/6 \\ & 1/8 & 3/8 & 3/8 & 1/8 & 0 \end{array}$$

Jelenítsük meg MATLAB® segítségével a megoldást:



3.4. ábra. A Brusselator modell megoldása változó lépésközű rácshálón

A 3.4. ábrán láthatjuk, hogy a változó lépésközzel kiegészített explicit Runge–Kutta módszer működik merev differenciálegyenletek megoldására. Megjegyezzük, hogy implicit Runge–Kutta módszerrel jobb közelítést tudnánk biztosítani, azonban ahhoz továbbra is nemlineáris egyenleteket kellene lépésenként megoldani, ami viszonylag költséges úton vezetne el a megoldásig.

4. fejezet

Összefoglalás

Elsődleges célunk a dolgozattal az volt, hogy bevezessük az olvasót a differenciálegyenletek megoldásait közelítő numerikus eljárások ismereteibe. A 2. fejezetben egy átfogó képet próbáltunk nyújtani a probléma jellegéről, majd ismertettünk a megoldáshoz alkalmas eljárási struktúrákat. Az explicit Euler módszer bevezetését követően, lépések közti lépcsők felvételével bevezettük a Runge–Kutta módszercsaládot, mely alapos vizsgálatával magasabb rendű módszereket voltunk képesek gyártani. Célunk volt továbbá, hogy az elméleti háttér mögött gyakorlati alkalmazásokba is betekintést nyújtsunk, ezért különféle MATLAB® kódok megírásával sokatmondó ábrákat is beékeltünk a sorok közé. Törekedtünk rávilágítani a differenciálegyenletek sokszínűségéből adódó nehézségekre is, melyekhez javítási módszereket vezettünk le. Így jutottunk el a 3. fejezetben látott rendfeltételkhez, becslésekhez, finommegoldásokhoz és további újdonságokhoz. A dolgozat végén pedig merev differenciálegyenletet tartalmazó feladat megoldásával is foglalkoztunk.

A numerikus modellezés világában nem létezik olyan, hogy valami tökéletesen pontos lenne, így a matematikának ez az ága számtalan új ismeretet rejt még magában, és megannyi megoldatlan problémára lesz képes választ adni. Csupán kitartónak és türelmesnek kell lennünk.

„A matematika bizonyos tekintetben mindig is az összekötő kapocs szerepét játszotta a különböző tudományok, valamint a tudomány és a művészet között. Meggyőződésem, hogy e tekintetben a matematikára a jövőben még fokozottabb szerep hárul.”

Rényi Alfréd

Irodalomjegyzék

[1] K. Sydsæter, P. Hammond, Matematika közgazdászoknak, Aula, 2000, 736. o.

[2] L. Euler, Institutionum Calculi Integralis, vol. XI., Volumen Primum, Opera Omnia, 1768

[3] K. Heun, Neue Methode zur approximativen Integration der Differentialgleichungen einer unabhägen Veränderlichen, Zeitschr. für Math. u. Phys., volt. 45, 1900, pp. 23–38.

[4] W. Kutta, Beitrag zur näherungsweisen Integration totaler Differentialgleichungen, Zeitschr. für Math. u. Phys., vol. 46, 1901, pp. 435–453.

[5] C. Runge, Ueber die numerische Auflösung von totaler Differentialgleichungen, Göttinger Nachr., 1905, pp. 252–257.

[6] J. C. Butcher, On Runge–Kutta processes of high order, Journal Australian Mathematical Society vol. IV, Part2, 1964, pp. 179–194.

[7] E. Harier, S. P. Nørsett, G. Wanner, Solving Ordinary Differential Equations I, Nonstiff Problems, Springer-Verlag Berlin Heidelberg, 1987

[8] J. C. Butcher, Numerical Methods for Ordinary Differential Equations, 2nd Edition, John Wiley Sons, 2008

[9] P. Albrecht, A new theoretical approach to Runge-Kutta methods, SIAM J. Numer. Anal., 24, 1987, pp. 391–406.

[10] P. Albrecht, The Runge–Kutta theory in a nutshell, SIAM J. Numer. Anal., 33, 1996, pp. 1712–1735.

[11] Randall J. LeVeque, Finite Difference Methods for Ordinary and Partial Differential Equations, SIAM, Philadelphia, 2007, pp. 255–258.

[12] K. Gustafsson, Control of Error and Convergence in ODE Solver, Licentiate Thesis Lund, 1992, pp. 22., Figure 2.2, pp. 30.

[13] Simon L. Péter, Közönséges differenciálegyenletek, Jegyzet, 2007, 1–6. o.

[14] Faragó István, Numerikus modellezés és közönséges differenciálegyenletek numerikus megoldási módszerei, Jegyzet, 2013

[15] http://www.hds.bme.hu/~cshos/matlab/matlab_hogyan/node36.html

[16] G. Söderlind, L. Jay, M. Calvo, Stiffness 1952-2012: Sixty years in search of a definition, BIT Numerical Mathematics 55 (2), 2015, pp. 531-558.

[17] D. Ketcheson, B. Seibold, D. Shirokoff, D. Zhou, DIRK Schemes with High Weak Stage Order, arXiv:1811.01285, 2018

[18] Uri A. Ascher, Numerical Methods for Evolutionary Differential Equations, SIAM, 2008, pp. 59-60.