
SZAKDOLGOZAT

Numerikus analízis alkalmazása az elektrokémiában: a Cottrell-kísérlet szimulációja

Témavezető:

Valkó Éva
Tanársegéd
Alkalmazott Analízis és Számításmatematikai Tanszék

Készítette:

Hojsza Kristóf
Matematika alapszakos hallgató



Eötvös Loránd Tudományegyetem
Természettudományi Kar
Budapest, 2019

Köszönetnyilvánítás

A szakdolgozatom elején szeretnék köszönetet mondani témavezetőmnek, Valkó Évának, egyrészt, mert megszerettette velem a Differenciálegyenletek és Parciális differenciálegyenletek tárgyakat, másrészt, hogy elvállalta a témavezetői teendőket és egy nagyon érdekes témán dolgozhattam. Hálás vagyok a hasznos konzultációkért, türelméért, építőjelleű kritikáiért és hogy bármikor számíthattam rá, így a szakdolgozat elnyerhette jelenlegi formáját.

Külön köszönettel tartozom még:

Vesztergom Somának, aki nagy segítségemre volt az elektrokémiai ismeretek megértésében és feldolgozásában.

valamennyi oktatómnak az Alkalmazott Analízis és Számításmatematikai Tanszékről.
családomnak, barátaimnak és csoporttársaimnak, akik az évek alatt támogattak.

Végül, de nem utolsósorban szeretnék még köszönetet mondani Rácz Olgának, hogy életem egyik legnehezebb és leggyötrelmesebb időszakában – a gimnáziumi évek alatt – hitt nekem és hitt bennem, biztatott, hogy bármilyen hátrányból is indulok, a szorgalmammal és kitartásommal el tudom érni, amit szeretnék.

Tartalomjegyzék

1. Bevezetés. Kémiai háttér	1
1.1. Elektrokémiai fémleválasztás; a Cottrell-kísérlet	1
1.2. Motiváció	2
2. Matematikai modell	3
2.1. Differenciálegyenletek	3
2.2. Fejezetek a komplex függvénytanból	5
2.2.1. Laplace-transzformáció	5
2.2.2. A generátorfüggvény deriváltjainak Laplace-transzformáltja	6
2.2.3. Inverz Laplace-transzformáció	6
3. Numerikus modell: Numerikus módszerek	8
3.1. Motiváció	8
3.2. Rácslábok és rácsfüggvények	9
3.3. Numerikus módszerekkel kapcsolatos alapfogalmak	10
3.3.1. Numerikus módszerek konvergenciája és konzisztenciája	10
3.3.2. Stabilitás, A-stabilitás	10
3.4. Euler-módszerek	11
3.4.1. Az explicit Euler-módszer	11
3.4.2. Az implicit Euler-módszer	12
3.5. Crank-Nicolson-módszer	13
3.6. A Θ -séma	15
3.7. A Runge-Kutta módszer	15
3.8. Lineáris többlépéses módszerek	18
4. Számítógépes modell: Cottrell-kísérlet	21
4.1. Az analitikus megoldás	21
4.2. A kísérlet szimulációja	25
4.2.1. A numerikus megoldás alapja	25
4.2.2. A szimuláció paraméterei	27
4.2.3. Az explicit Euler-módszer	28
4.2.4. Az implicit Euler-módszer	29
4.2.5. Crank-Nicolson-módszer	30
4.2.6. Runge-Kutta típusú módszerek	32
4.2.7. Lineáris többlépéses módszerek	34
4.3. A módszerek összehasonlítása	37
5. Összefoglalás	42
6. A szakdolgozatban szereplő Matlab kódok	43
Irodalomjegyzék	61
7. Eredetiségi nyilatkozat	62

1. Bevezetés. Kémiai háttér

Szakdolgozatom tárgya egy elektrokémiai kísérlet, az ún. Cottrell-kísérlet szimulációja többféle numerikus módszer felhasználásával.

Elektrokémiának nevezzük a fizikai kémia azon ágát, amely olyan rendszerekkel foglalkozik, amelyek viselkedését elektromos és kémiai kölcsönhatások egyaránt meghatározzák. Egy gyakran alkalmazott felosztás szerint az elektrokémiai kísérletek során vagy elektromos energiát alakítunk át kémiai energiává (és ún. elektrolízist végzünk), vagy pedig kémiai energiából igyekszünk elektromos energiát előállítani (ahogyan az például az elektrokémiai áramforrásokban is történik).

Bár a fenti felosztás se nem szükségszerű, se nem teljes, mégis elmondható, hogy szakdolgozatom elsősorban az első témakörhöz, az elektrolízis témaköréhez kapcsolódik. Egy elektrolízis során – legyen szó akár laboratóriumi kísérletről, akár ipari technológiáról – az a célunk, hogy villamos áram felhasználásával valamilyen (hasznos) kémiai változást idézzünk elő. Ez történik például akkor, amikor víz elektrolízisével hidrogén gázt állítunk elő (amelyet a későbbiekben energiaforrásként felhasználhatunk); ez az alapja az ún. klóralkáli iparnak, amellyel a vegy- és textilipar számára egyaránt fontos klórgáz előállítását végezzük; és ilyen pl. az elektrokémiai fémleválasztás folyamata is, amelynek során valamilyen fémsó oldatából (jellemzően bevonatként) tiszta fémeket állítunk elő. Utóbbi témakör (a fémleválasztás) különösen fontos szakdolgozatom szempontjából.

1.1. Elektrokémiai fémleválasztás; a Cottrell-kísérlet

Egy fémleválasztási kísérlet során lényegében az történik, hogy egy (elektromosan vezető, és jellemzően szilárd) hordozót (elektrodát) olyan oldattal hozunk kontaktusba, amely valamilyen oldott fémiont tartalmaz. Az elektródra (egy másik, a rendszerben elhelyezett ellenelektróddal szemben) negatív elektromos potenciált kapcsolva elérhetjük, hogy az oldat fémionjai, amint a hordozó felületét elérik, ott semlegesítődjenek, és szilárd fémbevonat formájában kiváljanak.

Eközben természetesen áramot mérhetünk, amelynek értéke arányos lesz a fémleválás reakciósebességével. Például a rézleválasztás folyamata során az oldat réz(II)-ionjai (Cu^{2+}) két elektron (e^-) felvételével fémréz (Cu) formába kerülnek,



amely reakció közben a rendszeren mólonként $2F$ ($F = 96485,3 \text{ C mol}^{-1}$ az ún. Faraday-állandó) töltés halad át.

E folyamat közben feltételezhetjük, hogy minden olyan rézion, amely a felületet eléri, ott egyből fémrézzé alakul; a rézionok koncentrációja az oldat/fém határfelület közvetlen közelében tehát zérus. Mivel az oldatba merülő fém a rézionokat folyamatosan fogyasztja, az oldat fémfelülethez közeledő része is rézionokban folyamatosan szegényedik. Bizonyos kísérleti körülmények között – ilyen az ún. Cottrell-kísérlet is – kimondható, hogy a koncentráció időbeli megváltozását a

$$\frac{\partial c}{\partial t} = D \frac{\partial^2 c}{\partial x^2} \quad (1.1.2)$$

parciális differenciálegyenlettel írhatjuk le, amelyben D a fémionok diffúziós együtthatója. (Az egyenletről és analitikus megoldásáról később, a 4. fejezetben még lesz szó.)

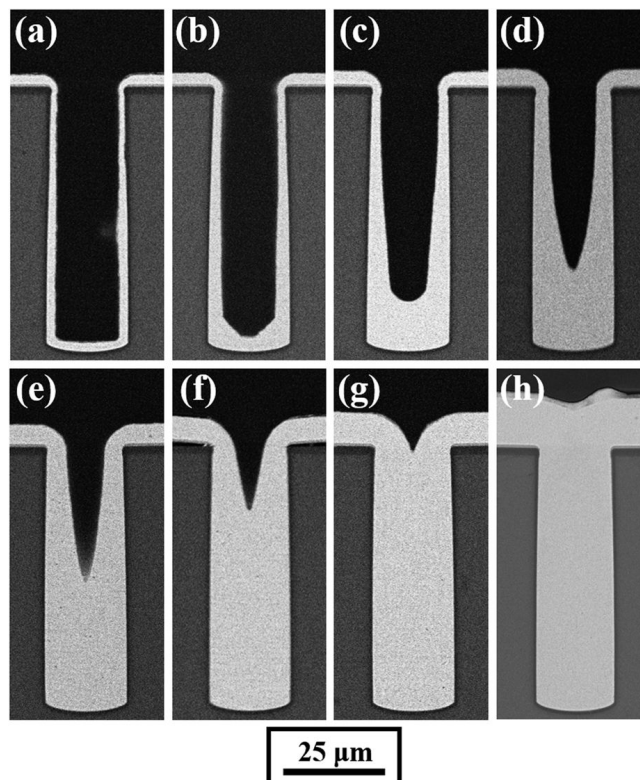
Megmutatható, hogy az ilyen rendszerekben a mérhető j áramsűrűség (a felületegységre vonatkoztatott áram) időfüggését az ún. Cottrell-egyenlettel írhatjuk le:

$$j = nFc^\infty \sqrt{\frac{D}{\pi t}}, \quad (1.1.3)$$

ahol n a reakcióra vonatkozó elektronszám (az 1.1.1 egyenletben ez 2), és c^∞ a tömbfázisbeli (az elektródtól végtelenül távol lévő, és ezért a kísérlet során meg nem változó) koncentráció. [3], [4], [5].

1.2. Motiváció

Az elektrokémiai fémleválasztás egy régóta vizsgált, és meglehetősen nagy kutatási hagyománnyal rendelkező kérdéskör, amely számos technológia alapját jelenti. Újabban az elektrokémiai fémleválasztás egyre nagyobb szerepet kap a mikroelektronikai iparban is, ahol a félvezető alapú integrált áramkörök fémes részeit elektrokémiai leválasztással készítik el [2]. Ez gyakran igen finoman mintázott (néhány 10 nm szélességű árkokat tartalmazó) felületek kompakt fémréteggel (hagyományosan rézzel, újabban kobalttal vagy akár nikkellel) történő bevonását jelenti.



1. ábra. Illusztráció a jól működő fémleválasztáshoz.

Egy jól működő technológia esetén a leválasztott fémréteg a leválasztás során először az árkokat és a kisebb (a szilíciumlapok közötti átvezetések biztosító) lyukakat kell, hogy kitöltse, majd a rétegnek egyenletesen kell növekednie egy bizonyos vastagságig (lásd 1. ábra¹). Ez a „szelektív leválás” jellemzően a leválasztó fürdő megfelelő adalékolásával érhető el; ilyenkor a fémleválasztás sebességét az adalékanyagok lokálisan megváltoztatják.

E többdimenziós, és kémiai szempontból sem egyszerű probléma analitikus kezelése sajnos lehetetlenül bonyolult feladat lenne – egyszerű analitikus megoldások, mint például a Cottrell-egyenlet, pedig bizonyosan nem elérhetők. A fémleválasztási technológiák elméleti kezelése során ezért elsősorban numerikus módszerek használatára hagyatkozunk: e ponton tehát kimondható, hogy a jó numerikus módszerek megtalálása kulcskérdése lehet a sikeres technológiafejlesztésnek is. A numerikus módszerek tesztelését viszont nyilvánvalóan nem érdemes bonyolult kérdésekkel kezdeni: szakdolgozatomban tehát ahhoz a célszerű megoldáshoz folyamodtam, hogy a különböző numerikus módszereket egy olyan rendszer (éppen a Cottrell-kísérlet) vizsgálatával hasonlítsam össze, ahol az analitikus megoldás is elérhető, és így a módszerek hatékonysága közvetlenül is mérhető. Remélhető, hogy az egyszerű és ellenőrizhető körülmények között jobban működő numerikus módszerek majd a bonyolultabb (már technológiai szempontból is releváns) problémák vizsgálatakor is hatékonyak bizonyulnak.

¹Az ábra forrása: <https://www.nature.com/articles/srep46639>

2. Matematikai modell

Ebben a fejezetben bemutatom a szükséges matematikai alapfogalmakat. Mivel a szakdolgozatnak nem célja a differenciálegyenletek explicit megoldási módszereinek bemutatása, illetve a szükségtelen hosszúság elkerülése is szempont volt, ezért ezeket nem részletezem. Azonban egy rövid bevezető erejéig itt megemlítem a differenciálegyenletekkel kapcsolatos alapvető ismeretek, tételek és összefüggések, amely ismeretekre a későbbiekben támaszkodni fogok. A bevezető után a Laplace-transzformációról és alkalmazásáról lesz szó, aminek fontos szerepe lesz az analitikus megoldás levezetésében. A differenciálegyenletek fejezetnél a [16], [17], [18] forrásokból építkeztem, a Laplace-transzformáció alfejezetekhez [19], [20] szakirodalmakat használtam fel.

2.1. Differenciálegyenletek

2.1.1. Definíció. (*n -edrendű közönséges differenciálegyenlet*) Legyen $F : \mathbb{R}^{n+2} \rightarrow \mathbb{R}$ folytonos függvény. $F(t, x(t), \dot{x}(t), \ddot{x}(t), \dots, x^{(n)}(t)) = 0$ n -edrendű közönséges differenciálegyenlet, ahol $x : \mathbb{R} \rightarrow \mathbb{R}$ az ismeretlen függvény

2.1.2. Definíció. (*Elsőrendű explicit differenciálegyenlet*) Legyen $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ folytonos függvény. Az $\dot{x}(t) = f(t, x(t))$ elsőrendű explicit közönséges differenciálegyenlet, ahol $x : \mathbb{R} \rightarrow \mathbb{R}$ folytonos függvény $x = x(t)$ az ismeretlen függvény és $t \in \mathbb{R}$.

2.1.3. Definíció. (*Cauchy-feladat*) Legyen $f : \mathbb{R}^{n+1} \rightarrow \mathbb{R}^n$ függvény, $T \subset \mathbb{R}^{n+1}$ tartomány. Az

$$\left. \begin{aligned} \dot{x} &= f(t, x) \\ x(0) &= x_0 \end{aligned} \right\}$$

feladatot Cauchy-feladatnak, vagy más néven kezdetiérték-problémának nevezzük, ahol $x_0 \in \mathbb{R}^n$ adott vektor. Az $x(0) = x_0$ adott vektor a megfelelő kezdeti feltétel.

2.1.4. Definíció. (*Cauchy-feladat megoldása*) Legyen $T \subset \mathbb{R}^{n+1}$ összefüggő nyílt tartomány és $I \subset \mathbb{R}$ nyílt intervallum, továbbá legyen $f : T \rightarrow \mathbb{R}^n$ folytonos függvény. Ha az $x : I \rightarrow \mathbb{R}^n$ differenciálható függvényre teljesülnek az alábbiak:

- $(t, x(t)) \in T$ minden $t \in I$ esetén,
- $\dot{x}(t) = f(t, x(t))$ minden $t \in I$ esetén,
- $x(t_0) = p_0$, ahol $(t_0, p_0) \in T$,

akkor az $x(t)$ függvényt az I intervallumon az f jobboldali explicit elsőrendű közönséges differenciálegyenlet(rendszer) megoldásának nevezzük az $x(t_0) = p_0$ kezdeti feltétel esetén.

Az alapfeladat és az alapfogalmak definiálása után most vizsgáljuk az alábbi három tulajdonságot:

- Mikor létezik egy differenciálegyenletnek megoldása? (Egzisztencia)
- Milyen feltételek mellett lesz egyértelmű a megoldás? (Unicitás)
- A megoldás folytonosan függ-e az adatoktól? (Stabilitás)

Létezés

2.1.1. Állítás. (*Peano*) Ha az $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ folytonos függvény, akkor az $\dot{x}(t) = f(t, x(t))$ differenciálegyenletnek bármely kezdeti feltétellel van megoldása.

Egyértelműség

Lemma. (*kis Gronwall-lemma*) Legyen $u : [a, b] \rightarrow \mathbb{R}$ differenciálható és tegyük fel, hogy $\dot{u}(t) \leq Ku(t)$ minden $t \in [a, b]$. Ekkor $u(t) \leq u(a)e^{K(t-a)}$ minden $t \in [a, b]$.

2.1.5. Definíció. (Lipschitz-feltétel²) Azt mondjuk, hogy egy függvény a második változójában Lipschitzes, ha $\|f(t, x_1) - f(t, x_2)\| \leq L \|x_1 - x_2\| \quad \forall (t, x_1), (t, x_2) \in T$, ahol az L nemnegatív állandó az ún. Lipschitz konstans.

2.1.1. Tétel. (Picard-Lindelöf) Adott az $\dot{x}(t) = f(t, x(t))$ közönséges differenciálegyenlet. Ha az $f(t, x(t))$ függvény a második változójában teljesíti a Lipschitz-feltételt, akkor a megoldás globálisan egyértelmű.

Folytonos függés

Legyen $T \subset \mathbb{R}^{n+1}$ tartomány, és jelölje \mathcal{F} a T halmazon folytonos, második változójában Lipschitz tulajdonságú függvények halmazát. A fent definiált kezdetiérték feladat bemenetei f függvény és (t_0, p_0) pont, míg a kimenete $\Phi_f(t, t_0, p_0)$ a megoldás, és annak maximális értelmezési intervalluma $I_f(t_0, p_0)$. A kimeneti adatok bemeneti adatoktól való függését három lépésben igazoljuk:

1. lépés: a kimeneti adatok folytonosan függenek p_0 ponttól.

2.1.2. Tétel. (Peano egyenlőtlenség) Tegyük fel, hogy $f \in \mathcal{F}$ függvény a T halmazon egyenletesen Lipschitz tulajdonságú a második változójában. Ha $\delta > 0$ olyan, amelyhez létezik olyan I nyílt intervallum, hogy $I \subset I(t, q)$, minden $|q - p_0| < \delta$ esetén, akkor minden $t \in I$ számra:

$$|\Phi_f(t, t_0, p_0) - \Phi_f(t, t_0, q)| \leq |p_0 - q| \cdot e^{L|t-t_0|}.$$

Lemma. Legyen $K \subset T$ kompakt halmaz. Ekkor bármely $f \in \mathcal{F}$ függvény az egész K halmazon egyenletesen Lipschitz tulajdonságú a második változójában

2.1.3. Tétel. (folytonos függés p_0 -tól) Legyen $f \in \mathcal{F}$, ekkor minden $(t_0, p_0) \in T$ ponthoz és minden $I \subset I(t_0, p_0)$ kompakt intervallumhoz létezik olyan $\delta > 0$ és $L > 0$, hogy minden q pontra, amelyre $(t_0, q) \in T$ és $|q - p_0| < \delta$, fennállnak a következők:

- $I \subset I(t_0, q)$
- $|\Phi_f(t, t_0, p_0) - \Phi_f(t, t_0, q)| \leq |p_0 - q| \cdot e^{L|t-t_0|} \quad t \in I.$

2. lépés: a megoldás értéke folytonosan függ az összes bemeneti adattól a különböző bemeneti adatokhoz tartozó megoldások közös értelmezési intervallumán.

Legyen $f \in \mathcal{F}$ és $(t_0, p_0) \in T$ rögzített, $I \subset I(t_0, p_0)$ kompakt intervallum, amely belsejében tartalmazza a t_0 pontot. Továbbá legyen $\Gamma \subset T$ a $\{(t, \Phi_f(t, t_0, p_0)) : t \in I\}$ görbe kompakt környezete és jelölje $\|g\|_\Gamma = \max_\Gamma |g|$, ha $g \in \mathcal{F}$.

2.1.4. Tétel. (Általános Peano egyenlőtlenség) Létezik olyan $L > 0$, hogy minden $(\tau, q) \in \Gamma$ és $g \in \mathcal{F}$ esetén, amelyekre $I \subset I_g(\tau, q)$, valamint $\{(t, \Phi_f(t, \tau, q)) : t \in I\} \subset \Gamma$, fennáll:

$$|\Phi_f(t, t_0, p_0) - \Phi_f(t, t_0, q)| \leq \left((\|f\|_\Gamma + \|f - g\|_\Gamma) |\tau - t_0| + |p_0 - q| \right) \cdot e^{L|t-t_0|} + \frac{\|f - g\|_\Gamma}{L} (e^{L|t-t_0|} - 1).$$

3. lépés: A kimeneti adatok folytonosan függenek az összes bemeneti adattól.

2.1.5. Tétel. (Folytonos függés) Legyen $f \in \mathcal{F}$. Ekkor minden $(t_0, p_0) \in T$ ponthoz, minden $I \subset I(t_0, p_0)$ kompakt intervallumhoz és minden $\epsilon > 0$ számhoz, melyre $\Gamma_\epsilon \subset T$, létezik olyan $\delta > 0$ és $L > 0$, hogy minden q pontra, melyre $(t_0, q) \in T$, $|\tau - t_0| < \delta$, $|q - p_0| < \delta$, $g \in \mathcal{F}$, és $\|g - f\|_{\Gamma_\epsilon} < \delta$ esetén fennállnak az alábbiak:

- $I \subset I_g(\tau, q)$
- fennáll az általánosított Peano egyenlőtlenség,

ahol $\Gamma_\epsilon = \{(t, p) \in T : t \in I, |p - \Phi(t, t_0, p_0)| \leq \epsilon\} \subset A$.

²Lipschitz-feltétel egy dimenzióban: Az f függvény teljesíti a Lipschitz-feltételt, ha \exists olyan $L > 0 : |f(x) - f(y)| \leq L |x - y|$, ahol $x, y \in D(f)$.

2.2. Fejezetek a komplex függvénytanból

A Laplace-transzformáció egyik nagyon fontos gyakorlati alkalmazása állandó együtthatójú differenciálegyenletek és egyenletrendszerek megoldása. Tekintsük az egyik ilyen nevezetesebb típusú egyenletet, a másodrendű differenciálegyenletet:

$$a \cdot y'' + b \cdot y' + c \cdot y = f(t)$$

ahol a, b, c adott konstansok, $f(t)$ adott függvény és $y(t)$ az ismeretlen függvény. A megoldási elv a következő: "leegyszerűsítjük a problémát", hiszen differenciálegyenlet helyett csupán egy algebrai egyenletet oldunk meg. Differenciálegyenlet-rendszer esetén algebrai lineáris egyenletrendszerre vezetjük vissza a feladatunkat. A fejezet végén egy konkrét példán keresztül szemléltetem a differenciálegyenlet megoldását Laplace-transzformációval.

2.2.1. Laplace-transzformáció

Legyen t egy valós változó, és f ennek valamely valós vagy komplex értékű függvénye.³

2.2.1. Definíció. (Laplace-transzformáció) A Laplace-transzformáció az f valós változójú - valós vagy komplex értékű - függvényhez az alábbi F komplex változós függvényt rendeli:

$$F(s) := \int_0^{\infty} f(t)e^{-st} dt$$

Az ezen a módon előállított komplex s változótól függő $F(s)$ függvényt nevezzük az $f(t)$ valós változós függvény Laplace-transzformáltjának.⁴

Megjegyzés. Egyszerűbben a következőképp jelöljük az f függvény Laplace-transzformáltját:
 $L[f] = F$

A definíció után a legfontosabb, hogy áttekintsük, milyen feltételeknek kell fennállnia ahhoz, hogy egy f függvénynek létezzen Laplace-transzformáltja. Ehhez annak kell teljesülnie, hogy az s komplex paramétertől függő improprius integrál konvergens legyen, ahhoz pedig a következő három feltételnek kell teljesülnie:

1. Az f függvénynek egy véges hosszúságú intervallumon belül legfeljebb véges számú ugrása legyen.
2. bármely $t_0 < t_1 \leq t \leq t_2$ intervallumra vonatkozó integrálnak korlátosnak kell lennie.
3. teljesüljön az alábbi két határérték reláció, ahol $\alpha < \sigma$, $\sigma > 0$, $n > 0$:

- $\lim_{t \rightarrow \infty} t^n \cdot e^{-\sigma t} = 0$
- $\lim_{t \rightarrow \infty} e^{\alpha t} \cdot e^{-\sigma t} = 0$

2.2.1. Állítás. A Laplace-transzformáció lineáris, tehát az összeget lehet tagonként transzformálni és a konstans szorzó a transzformáció során kiemelhető:

$$L \left[\sum_{i=0}^n c_i f_i(t) \right] = c_i \sum_{i=0}^n L[f_i(t)]$$

³Általános jelölés: a t változóval az időt, x, y változókkal a térkoordinátákat szokás jelölni, hiszen valamilyen természettudományi modellt írunk le. Olyan jelenségeket vizsgálunk, amelyben f időfüggvény értelmezési tartománya adott $t_0 \in \mathbb{R}$ esetén a $t \geq t_0$ félegyenes, tehát ami egy t_0 időpontban kezdődő folyamatot, változást jellemez.

⁴az f függvényt szokás eredeti függvénynek vagy generátorfüggvénynek nevezni.

$f(t)$	$F(s)$	$f(t)$	$F(s)$	$f(t)$	$F(s)$
$c \in \mathbb{R}$	$\frac{c}{s}$	$\frac{1}{\sqrt{t}}$	$\frac{\sqrt{\pi}}{\sqrt{s}}$	$\sin(\alpha t)$	$\frac{\alpha}{s^2 + \alpha^2}$
$t^n \quad n \in \mathbb{N}$	$\frac{n!}{s^{n+1}}$	\sqrt{t}	$\frac{\sqrt{\pi}}{2 \cdot s^{\frac{3}{2}}}$	$\cos(\alpha t)$	$\frac{s}{s^2 + \alpha^2}$
e^{at}	$\frac{1}{s-a}$	$\ln(t)$	$-\frac{1}{s}(C + \ln(s))$	$sh(\alpha t)$	$\frac{\alpha}{s^2 - \alpha^2}$
$t \cdot e^{at}$	$\frac{1}{(s-a)^2}$	$\frac{1}{\sqrt{t}} \cdot \ln(t)$	$-\sqrt{\frac{\pi}{s}}(C \cdot \ln(4) + \ln(s))$	$ch(\alpha t)$	$\frac{s}{s^2 - \alpha^2}$

1. táblázat. Nevezetes Laplace-transzformáltak

2.2.2. A generátorfüggvény deriváltjainak Laplace-transzformáltja

A következőben arra nézünk eljárást, hogy egy generátorfüggvény deriváltjának hogyan állítsuk elő a Laplace-transzformáltját. Alkalmazzuk a deriváltra a definíciót:

$$\begin{aligned}
 L[f'(t)] &= \int_0^{\infty} f'(t) \cdot e^{-st} dt = [f(t) \cdot e^{-st}]_0^{\infty} - \int_0^{\infty} f(t)(-s)e^{-st} dt = -f(0) + s \cdot \int_0^{\infty} f(t)e^{-st} dt = \\
 &= s \cdot \underbrace{L[f(t)]}_{F(s)} - f(0).
 \end{aligned}$$

Tehát az $L[f'(t)] = s \cdot L[f(t)] - f(0)$ kifejezéssel megadható az eredeti függvény deriváltjának a Laplace-transzformáltja. Vegyük észre, hogy az $f(0) = 0$ esetben $f'(t)$ Laplace-transzformáltjának meghatározásához $L[f(t)]$ függvényt csak s -sel kell megszoroznunk. Most vizsgáljuk meg a második derivált esetén mi történik:

$$L[f''(t)] = sL[f'(t)] - f'(0) = s(sL[f(t)] - f(0)) - f'(0) = s^2L[f(t)] - sf(0) - f'(0).$$

Nyilvánvalóan $f(0) = f'(0) = 0$ esetben itt egy s^2 -tel való szorzásról van szó. Összehasonlítva az első és második deriváltra vonatkozó összefüggéseket sejteti magát a következő formula:

$$L[f^{(n)}(t)] = s^n L[f(t)] - s^{n-1}f(0) - s^{n-2}f'(0) - s^{n-3}f''(0) - \dots - f^{(n-1)}(0)$$

Könnyen belátható teljes indukcióval, hogy a fenti összefüggés fennáll tetszőleges $n \in \mathbb{N}^+$ esetén, ami egyértelműen kényelmesebb és rövidebb, mint definíció szerint improprius integrálokat meghatározni.

2.2.3. Inverz Laplace-transzformáció

Eddig arra néztünk módszereket, hogyan határozhatjuk meg egy generátorfüggvény és deriváltjainak Laplace-transzformáltját, azonban a feladat megfordítása is ugyanennyire hasznos, azaz most azt mutatom be, hogyan határozhatjuk meg egy Laplace-transzformált eredeti függvényét.

2.2.2. Definíció. (Konvolúció) Adott f_1 és f_2 generátorfüggvények. E két függvény konvolúcióján (jele: $f_1 * f_2$) az alábbi összefüggést értelmezzük:

$$(f_1 * f_2)(t) = \int_0^t f_1(s) \cdot f_2(t-s) ds$$

2.2.2. Állítás. Az alábbi integrálformula segítségével határozhatjuk meg egy adott F Laplace-transzformált f generátorfüggvényét:

$$f(t) = \frac{1}{2\pi i} \int_{\sigma-i\infty}^{\sigma+i\infty} F(s) \cdot e^{st} ds \quad t \geq 0.$$

Az $L[f] = F$ kifejezés jelöli a f generátorfüggvény Laplace-transzformáltját. Jelölje az inverz kapcsolatot $L^{-1}[F] = f$

2.2.1. Tétel. (Konvolúciótétel) Ha f transzformáltja F , g transzformáltja G , akkor:

$$L^{-1}[F(s) \cdot G(s)] = (f * g)(t) = \int_0^t f(s) \cdot g(t-s) ds$$

2.2.2. Tétel. (A generátor függvény deriválási tétele)

$$L^{-1}[s \cdot F(s)] = L^{-1}\left[L[f'(t)] + f(0)\right]$$

2.2.3. Tétel. (Transzformált deriválási tétele)

$$L^{-1}[F'(t)] = -t \cdot f(t)$$

Végezetül jöjjön egy hasznos példa az eddig tárgyalt ismeretekre bemutatására.

2.2.1. Példa. Oldjuk meg az alábbi differenciálegyenletet: $y'' + y = 4 \cdot e^t$, $y(0) = 4$, $y'(0) = -3$!

Tagonként előállítva a Laplace-transzformáltat:

$$s^2 \cdot Y(s) - s \cdot y(0) - y'(0) + Y(s) = 4 \cdot \frac{1}{s-1}$$

Felhasználva a feladatban szereplő kezdeti feltételeket:

$$s^2 \cdot Y(s) - 4 \cdot s + 3 + Y(s) = \frac{4}{s-1}$$

Az előző átalakításokkal az eredeti feladatunkat visszavezettük egy egyszerű algebrai egyenletre $Y(s)$ -re nézve, amelyet $Y(s)$ -re rendezve:

$$Y(s) = \frac{4}{(s-1)(s^2+1)} + \frac{4s-3}{s^2+1}$$

Alkalmazva a parciális törtekre bontást:

$$Y(s) = \frac{2s}{s^2+1} - \frac{5}{s^2+1} + \frac{2}{s-1}$$

A táblázatból leolvasva az inverz transzformáltakat, megkapjuk a differenciálegyenlet megoldását:

$$y(t) = 2 \cos(t) - 5 \sin(t) + 2e^t$$

Ahogy az a megoldási módszerrel látható, sokkal egyszerűbben állítottuk elő a megoldást, mintha az általános megoldási módszerrel számoltuk volna ki a homogén egyenlet általános megoldása és az inhomogén egyenlet partikuláris megoldása segítségével. ♦

3. Numerikus modell: Numerikus módszerek

Ebben a szakaszban a numerikus megoldási módszereket fogom bemutatni. Egy kisebb motivációs rész után a numerikus analízis két jelentős ágával: az interpolációval és a kezdetiérték feladat numerikus megoldásával fogunk megismerkedni. Ehhez a fejezethez a [12], [13], [14] forrásokat használtam fel.

3.1. Motiváció

Mielőtt mélyebben megismerkednénk a különböző numerikus módszerekkel és alkalmazásaikkal, tekintsünk át rövidebben néhány problémakört, amelyek kihangsúlyozzák a numerikus analízis fontosságát. Először is nézzünk egy motiváló példát!

3.1.1. Példa. Oldjuk meg az alábbi kezdetiérték-feladatot!

$$\left. \begin{aligned} \dot{y}(t) &= y(t) \\ y(0) &= 1 \end{aligned} \right\}$$

$y(t)$ könnyen meghatározható akár közvetlenül is, $y(t) = e^t$. Tegyük fel, hogy mi például $t=0,5$ helyen vagyunk kíváncsiak a felvett értékre. Ekkor $y(0,5) = e^{0,5} = \sqrt{e}$. Mivel e irracionális szám, egy irracionális számból gyököt vonva szintén azt kapunk. Tehát ilyen esetben nem tudjuk meghatározni a pontos megoldást, kénytelenek vagyunk rá egy közelítést adni. ♦

A numerikus módszerek jelentőségét kihangsúlyozza, hogy sokszor nem tudjuk nélküle megoldani az adott problémát. Vannak olyan esetek is, amikor még ha meg is tudnánk oldani az adott feladatot numerikus módszerek nélkül, azonban mégis inkább azokat alkalmazzuk, például a költséghatékonyság miatt, vagy a hatalmas műveletigény elkerülése szempontjából.

Utóbbira példa lehet az egyenletrendszerek megoldása vagy a sajátértékszámítás. Nagy mátrixok esetén sokkal könnyebb egy közelítést adni, mint elemi módszerekkel meghatározni a megoldást. Egy példa az első esetre a numerikus integrálás, hiszen egy határozott integrál meghatározása során sok komplikáció léphet fel, tehát nem minden esetben tudjuk használni a Newton-Leibniz-formulát⁵. Ilyen eshetőség lehet például, hogy túl bonyolult előállítani az adott függvény primitív függvényét, vagy nem tudjuk⁶. Vagy olyan helyzetbe kerülünk, mint a motivációs példában. Esetleg az is előfordulhat, hogy nem ismerjük magát a függvényt, amit integrálni akarunk, erre egy magyarázat lehet, hogy csak kevés adatot ismerünk, például, hogy az adataink mérési eredményből származnak, azaz csak véges sok pont áll a rendelkezésünkre.

A következő alfejezetekben numerikus megoldási módszereket mutatok be a kezdetiérték-feladat témaköréből.

⁵Newton-Leibniz-formula: Tegyük fel, hogy f integrálható $[a, b]$ -ben. Abban az esetben, ha F folytonos $[a, b]$ -n, differenciálható (a, b) -n és $\forall x \in (a, b)$ -re teljesül, hogy $F'(x) = f(x)$, akkor az f függvény határozott integrálja:

$$\int_a^b f(x) dx = F(b) - F(a)$$

⁶Ilyenek függvények például: $\int e^{-\frac{x^2}{2}} dx$, $\int \frac{\sin(x)}{x} dx$, $\int \frac{1}{\log(x)} dx$, $\int \sin x^2 dx$

3.2. Rácshálók és rácsfüggvények

Először is ismertetem a rácshálók és rácsfüggvények fogalmát, hiszen ezek fognak alapul szolgálni a kezdetiérték-probléma numerikus megoldásához. A célunk egy olyan rácsfüggvény előállítását, amely jól közelíti a Cauchy-feladat megoldását. Legyen $h > 0$ adott szám és generáljunk rácsfüggvényt a következő módon a $[0, T]$ intervallumon:

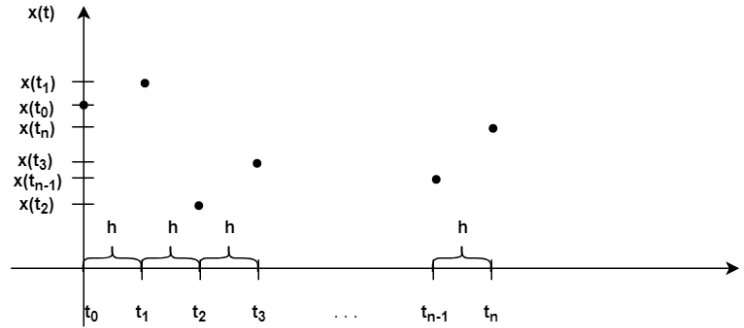
$$w_h := \{t_0 = 0 < h < 2h < 3h < \dots < Nh = T\}.$$

Jelöljük inentől ezeket a pontokat a következőképp:

$$t_n = nh \quad n = 0, 1, \dots, N \quad h = T/N,$$

melyek ekvidisztáns vagy más néven egyenközű rácshálót alkotnak és h pedig a lépésköz.

3.2.1. Definíció. (Rácsfüggvény) Az $y_h : w_h \rightarrow \mathbb{R}$ függvényt, azaz a rácsháló pontjaiból a valós számok halmazába történő leképezést a w_h rácsháléhoz tartozó rácsfüggvénynek nevezzük.

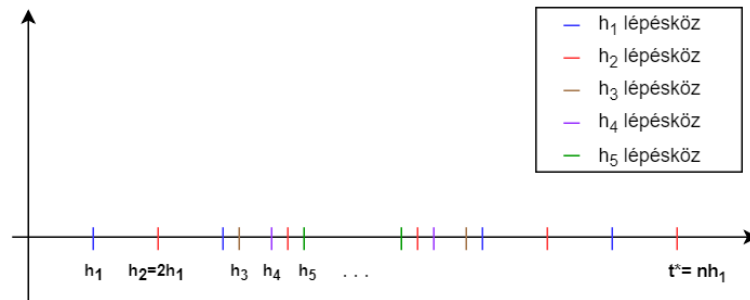


2. ábra. Rácsfüggvény

Olyan rácsfüggvényeket állítunk elő, amelyek jól közelítik a kezdetiérték-probléma pontos megoldását:

$$y_h(t_n) \approx u(t_n) \quad t_n \in w_h$$

Tekintsünk egy (w_h) rácsháló sorozatot, amelyre tegyük fel hogy teljesül, hogy van olyan $t^* \in w_h$ pont, amely minden h lépésköz esetén része a megfelelő w_h rácshálónak.



3. ábra. Példa rácsháló sorozatra 5 fajta lépésköz esetén

3.2.2. Definíció. (Jó közelítés) Azt mondjuk, hogy az (y_h) rácsfüggvényssorozat jól közelíti az $u(t)$ megoldást egy t^* rögzített pontban, ha

$$\lim_{h \rightarrow 0} |y_h(t^*) - u(t^*)| = 0.$$

3.3. Numerikus módszerekkel kapcsolatos alapfogalmak

A numerikus módszer ismertetése előtt tekintsük át a legfontosabb tulajdonságait, illetve hogy mik a főbb elvárásaink velük szemben. Az egyszerűség kedvéért vezessük be a közelítő rácsfüggvényre $y_n := y_n(t_n)$ jelölést. Jelölje y_n a feladat közelítő megoldását a t_n időpillanatban, míg $u(t)$ pedig a pontos megoldást.

3.3.1. Numerikus módszerek konvergenciája és konzisztenciája

Először is fontos a hiba mértékének a jellemzése, amelyet a következőképp definiálunk.

3.3.1. Definíció. (Globális hiba) Jelölje ezentúl e_n a globális hibát, amely a közelítés és a pontos megoldása különbsége: $e_n(h) = y_n - u(t^*)$, $nh = t^*$, $n = 0, 1, \dots, N$.

A numerikus módszerrel szemben az alapvető elvárásunk a következő:

$$\lim_{h \rightarrow 0} e_n(h) = 0. \quad (3.3.1)$$

3.3.2. Definíció. (Pontbeli konvergencia) A numerikus módszer konvergens a $t = t^*$ pontban, ha $h \rightarrow 0$ esetén a 3.3.1 egyenlőség igaz.

3.3.3. Definíció. (p -edrendű konvergencia) A numerikus módszer p -edrendben konvergens, ha $|y_n - u(t^*)| = \mathcal{O}(h^p)$, azaz

$$\lim_{\substack{h \rightarrow 0 \\ n \rightarrow \infty}} |y_n - u(t^*)| = 0$$

p -edrendben.

3.3.4. Definíció. (Konvergencia) A numerikus módszer konvergens (p -edrendben), ha $[0, t]$ intervallum minden pontjában az.

3.3.5. Definíció. (Konzisztencia rend) Azt mondjuk, hogy a numerikus módszer p -edrendben approximálja a pontos megoldást, ha módszer approximációs hibájára $< \mathcal{O}(h^p)$. Ekkor azt mondjuk, hogy numerikus módszer konzisztencia rendje p .

3.3.2. Stabilitás, A-stabilitás

Az egy lépéses módszerek egy újabb fontos tulajdonságát vizsgáljuk. Tekintsük az úgynevezett Dahlquist-féle tesztfeladatot:

$$\begin{cases} y'(t) &= \lambda y(t), \\ y(0) &= y_0, \end{cases}$$

ahol $\lambda \in \mathbb{C}^-$.

3.3.6. Definíció. (A-stabil) Azt mondjuk, hogy egy numerikus módszer A-stabil, ha $S = \{z \in \mathbb{C} : |R(z)| \leq 1\} \supseteq \mathbb{C}^-$, ahol S a stabilitási tartomány és $R(z)$ a módszerhez tartozó stabilitási függvény.

3.3.1. Tétel. Tegyük fel, hogy a numerikus módszer

- p -edrendben konzisztens,
- stabil.

Ekkor a numerikus módszer p -edrendben konvergens.

3.4. Euler-módszerek

Az előzőekben megismerkedtünk a rácsfüggvényekkel és a legfontosabb alapfogalmakkal. Most különböző numerikus módszereket ismertetek a rácsfüggvény előállítására.

3.4.1. Az explicit Euler-módszer

Tekintsük az előző fejezetben definiált Cauchy-feladatot:

$$\begin{aligned} u'(t) &= f(t, u(t)) \\ u(0) &= u_0. \end{aligned}$$

$u'(t_n) = f(t_n, u(t_n))$ minden n -re. Jelölje y_n a pontos megoldás közelítését a t_n időpillanatban, azaz $y_n \approx u(t_n)$. Ekkor a Cauchy-feladat: $u'(t_n) = f(t_n, u(t_n))$ minden $t \in I$ -re. Felhasználva a haladó véges differenciás approximációt $u'(t_n) \approx \frac{y_{n+1} - y_n}{h}$, míg a jobboldal esetén a következő közelítést alkalmazva: $f(t_n, y_n) \approx f(t_n, u(t_n))$:

$$\frac{y_{n+1} - y_n}{h} = f(t_n, y_n) \quad n = 0, 1, \dots, N-1$$

Rendezve y_{n+1} -re a következő összefüggést kapjuk:

3.4.1. Definíció. (*Explicit Euler-módszer*)⁷ Az alábbi összefüggést egy lépéses explicit Euler-módszernek nevezzük:

$$\begin{aligned} y_{n+1} &= y_n + hf(t_n, y_n) \\ y_0 &= u_0 \end{aligned} \tag{3.4.1}$$

Megjegyzés. Az explicit jelző arra utal, hogy a t_n pontbeli érték ismeretében a t_{n+1} pontbeli közelítés megadása egy behelyettesítéssel kiszámolható.

Tekintsük az explicit Euler-módszert, és rendezzük a globális hibaegyenletet a következőképpen: $y_n = e_n + u_n$. Ezt behelyettesítve az explicit Euler-módszert leíró képletbe:

$$\begin{aligned} \frac{(e_{n+1} + u_{n+1}) - (e_n + u_n)}{h} &= f(t_n, y_n) \\ \frac{e_{n+1} - e_n}{h} &= -\frac{u_{n+1} - u_n}{h} + f(t_n, e_n + u_n) = \underbrace{\left[-\frac{u_{n+1} - u_n}{h} + f(t_n, y_n) \right]}_{\Psi_n^{(1)}} + \underbrace{\left[f(t_n, e_n + u_n) - f(t_n, y_n) \right]}_{\Psi_n^{(2)}} \\ \frac{e_{n+1} - e_n}{h} &= \Psi_n^{(1)} + \Psi_n^{(2)} \quad , \text{ ahol } \Psi_n^{(1)} \text{ a lokális approximációs hiba.} \end{aligned}$$

3.4.1. Állítás. Az explicit Euler-módszer elsőrendben konzisztens.

Bizonyítás.

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

felhasználva, hogy $f(t_n, u(t_n)) = u'(t_n)$

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

Tehát az explicit Euler-módszer elsőrendben konzisztens. □

⁷Az 1768-as explicit Euler-módszer, vagy másnéven haladó Euler-módszer (forward Euler method).

Vizsgáljuk meg az explicit Euler-módszert stabilitás szempontjából:

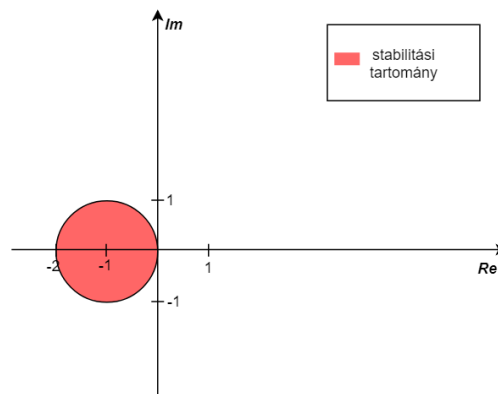
$$y_{n+1} = y_n + hf(t_n, y_n)$$

$$y_{n+1} = y_n + h\lambda y_n$$

bevezetve a $z := h\lambda$ jelölést:

$$y_{n+1} = \underbrace{(1 + z)}_{R(z)} y_n$$

Miután meghatároztuk $R(z)$ stabilitási függvényt, a 4. ábrán látható az explicit Euler-módszer stabilitási tartománya.



4. ábra. Az explicit Euler-módszer stabilitási tartománya

Mivel $\mathbb{C}^- \not\subseteq S$, az alábbi állítás mondható el:

3.4.2. Állítás. *Az explicit Euler-módszer nem A-stabil.*

3.4.3. Állítás. *Az explicit Euler-módszer 0-stabil.*

Mivel a módszer 0-stabil, illetve elsőrendben konzisztens, így a 3.3.1. tétel alapján az alábbi következményt kapjuk:

3.4.4. Állítás. *Az explicit Euler-módszer elsőrendben konvergens.*

3.4.2. Az implicit Euler-módszer

3.4.2. Definíció. (Az implicit Euler-módszer) A következőképp definiált módszert implicit Euler-módszernek nevezzük:

$$\begin{aligned} y_{n+1} &= y_n + hf(t_{n+1}, y_{n+1}) \\ y_0 &= u_0 \end{aligned} \tag{3.4.2}$$

Tekintsük át az implicit Euler-módszer legfontosabb tulajdonságait.

Megjegyzés. Az implicit tulajdonság jelentése, hogy itt már a t_{n+1} pontbeli közelítés meghatározása a t_n pontbeli érték felhasználásával nem olyan egyszerű, mint az explicit Euler módszer esetén, hiszen itt egy függvénybehelyettesítés helyett általában nemlineáris egyenlet megoldásával adható meg, tehát műveletigény szempontjából egyértelműen költségesebb ez a módszer.

3.4.5. Állítás. *Az implicit Euler-módszer elsőrendben konzisztens.*

Bizonyítás.

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

$$\begin{aligned} \Psi_n^{(1)} &= -\frac{u(t_{n+1}) - u(t_n)}{h} + f(t_{n+1}, u(t_{n+1})) = -\frac{u(t_{n+1}) - u(t_{n+1} - h)}{h} + \\ &+ f(t_{n+1}, u(t_{n+1})) = -\frac{u(t_{n+1}) + hu'(t_{n+1}) + \mathcal{O}(h^2) - u(t_{n+1})}{h} + u'(t_{n+1}) = \mathcal{O}(h) \end{aligned}$$

Vagyis az implicit Euler-módszer elsőrendben konzisztens. \square

Határozzuk meg az implicit Euler-módszer stabilitási függvényét!

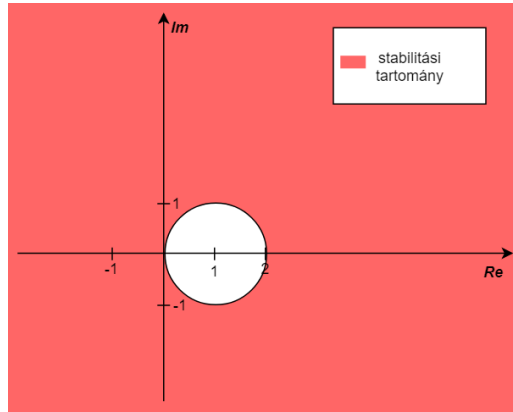
$$y_{n+1} = y_n + hf(t_{n+1}, y_{n+1})$$

$$y_{n+1} = y_n + h\lambda y_{n+1}$$

y_{n+1} -re rendezve és bevezetve a $z := h\lambda$ jelölést:

$$y_{n+1} = \underbrace{\frac{1}{1-z}}_{R(z)} y_n.$$

A 5. ábrán az implicit Euler-módszer stabilitási tartománya látható.



5. ábra. Az implicit Euler-módszer stabilitási tartománya

Mivel itt már $\mathbb{C}^- \subseteq S$, így teljesül az alábbi:

3.4.6. Állítás. *Az implicit Euler-módszer A-stabil.*

Az A-stabilitás és az elsőrendű konzisztencia a 3.3.1. tétel alapján szintén biztosítja az alábbi következményt:

3.4.7. Állítás. *Az implicit Euler-módszer elsőrendben konvergens.*

3.5. Crank-Nicolson-módszer

Állítsunk elő rácsgüggvényt úgy, hogy vegyük az explicit Euler-módszer és az implicit Euler-módszer számtani közepét:

$$\frac{y_{n+1} - y_n}{h} = \frac{1}{2} [f(t_n, y_n) + f(t_{n+1}, y_{n+1})]$$

3.5.1. Definíció. (Crank-Nicolson-módszer) Az így előállított formulát Crank-Nicolson- módszernek, vagy másnéven Trapéz-szabálynak nevezzük:

$$\boxed{\begin{aligned} y_{n+1} &= y_n + \frac{h}{2} [f(t_n, y_n) + f(t_{n+1}, y_{n+1})] \\ y_0 &= u_0 \end{aligned}} \quad (3.5.1)$$

Vizsgáljuk meg ezt a módszert is konzisztencia és stabilitás szempontjából.

3.5.1. Állítás. A Trapéz-módszer másodrendben konzisztens.

Bizonyítás.

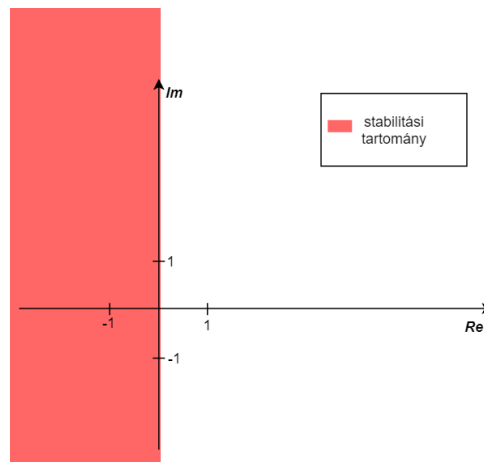
$$\begin{aligned} u(t_{n+1}) &= u(t_n + h) = u(t_n) + hu'(t_n) + \frac{h^2}{2}u''(t_n) + \mathcal{O}(h^3) \\ u'(t_{n+1}) &= u'(t_n + h) = u'(t_n) + hu''(t_n) + \mathcal{O}(h^2) \\ \Psi_n^{(1)} &= -\frac{u(t_{n+1}) - u(t_n)}{h} + \frac{1}{2}[f(t_n, u_n) + f(t_{n+1}, u_{n+1})] = -\frac{u(t_{n+1}) - u(t_n)}{h} + \\ &+ \frac{1}{2}[u'(t_n) + u'(t_{n+1})] = -\frac{u(t_n) + hu'(t_n) + \frac{h^2}{2}u''(t_n) + \mathcal{O}(h^3) - u(t_n)}{h} + \\ &+ \frac{1}{2}[u'(t_n) + u'(t_n) + hu''(t_n) + \mathcal{O}(h^2)] = \mathcal{O}(h^2) \end{aligned}$$

azaz, a Trapéz-módszer másodrendben konzisztens. \square

Meghatározva a Crank-Nicolson-módszer stabilitási függvényét:

$$\begin{aligned} y_{n+1} &= y_n + \frac{h}{2} [f(t_n, y_n) + f(t_{n+1}, y_{n+1})] \\ y_{n+1} &= y_n + \frac{h\lambda}{2} y_n + \frac{h\lambda}{2} y_{n+1} \\ y_{n+1} &= \underbrace{\frac{1 + z/2}{1 - z/2}}_{R(z)} y_n \end{aligned}$$

A 6. ábrán látható, hogy $\mathbb{C}^- \subseteq S$, így ennél a módszernél is elmondható az alábbi állítás.



6. ábra. A Crank-Nicolson-módszer stabilitási tartománya

3.5.2. Állítás. A Crank-Nicolson-módszer A-stabil.

A Trapéz-szabály A -stabilitása és a másodrendű konzisztenciája már egy magasabb konvergenciát tesz lehetővé, mint az eddigi módszerekénél:

3.5.3. Állítás. *A Crank-Nicolson-módszer másodrendben konvergens.*

3.6. A Θ -séma

A Crank-Nicolson-módszer másodrendű konvergenciája után felmerül a kérdés, hogy van-e más másodrendű egy lépéses módszer. A válasz a következő eljárást vizsgálva megkapható:

3.6.1. Definíció. (θ -séma) Legyen $\theta \in [0, 1]$ rögzített szám. Az alábbi eljárást θ -sémának nevezzük:

$$\begin{array}{lcl} y_{n+1} & = & y_n + h \left[(1 - \theta)f(t_n, y_n) + \theta f(t_{n+1}, y_{n+1}) \right] \\ y_0 & = & u_0 \end{array} \quad (3.6.1)$$

3.6.1. Állítás. *A Θ -séma csak $\Theta = 0,5$ esetén másodrendben konzisztens, más Θ esetén csak elsőrendű a konzisztencia.*

Bizonyítás.

$$\begin{aligned} u(t_{n+1}) &= u(t_n + h) = u(t_n) + hu'(t_n) + \frac{h^2}{2}u''(t_n) + \frac{h^3}{3!}u'''(t_n) + \mathcal{O}(h^4) \\ u'(t_{n+1}) &= u'(t_n + h) = u'(t_n) + hu''(t_n) + \frac{h^2}{2}u'''(t_n) + \mathcal{O}(h^3) \\ \Psi_n^{(1)} &= -\frac{u(t_{n+1}) - u(t_n)}{h} + (1 - \Theta)f(t_n, u_n) + \Theta f(t_{n+1}, u_{n+1}) = \\ &= -\frac{u(t_n) + hu'(t_n) + \frac{h^2}{2}u''(t_n) + \frac{h^3}{6}u'''(t_n) + \mathcal{O}(h^4) - u(t_n)}{h} + \\ &\quad + u'(t_n) - \Theta u'(t_n) + \Theta \left(u'(t_n) + hu''(t_n) + \frac{h^2}{2}u'''(t_n) + \mathcal{O}(h^3) \right) = \\ &= -\frac{h}{2}u''(t_n) - \frac{h^2}{6}u'''(t_n) + \mathcal{O}(h^3) + \Theta hu''(t_n) + \Theta \frac{h^2}{2}u'''(t_n) \end{aligned}$$

Ahonnán következik, hogy a másodrendű konzisztenciához az alábbi egyenletnek kell eleget tenni:

$$-\frac{h}{2} + \Theta h = 0,$$

ami csak akkor fog teljesülni, ha $\Theta = 0,5$. Tehát csak ebben az egy megválasztásban teljesül a másodrendű konzisztencia, a többi esetben csak elsőrendű konzisztencia biztosított, magasabbrendű konzisztencia könnyen láthatóan nem érhető el. \square

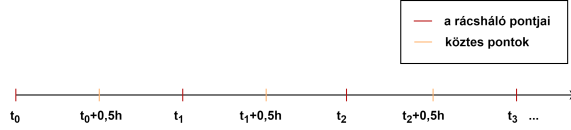
Megjegyzés. Vegyük észre, hogy a megfelelő megválasztásokkal az előbb definiált módszereket kapjuk vissza: $\theta = 0, 1, \frac{1}{2}$ rendre az explicit, implicit Euler-módszer és trapéz módszer.

3.7. A Runge-Kutta módszer

Az előző alfejezetekben láthattuk, hogy kezdetiérték problémára az eddig ismertetett egy lépéses módszerek legfeljebb másodrendű konvergenciát biztosítanak. Az ilyen módszerek előállítása kétféleképpen lehetséges, a következő kettő alfejezet ezzel foglalkozik. Most az alábbi kérdésre keressük a választ: gyártható-e magasabb rendben konvergens explicit módszer? Származtassunk egy ilyen eljárást, az explicit Euler-módszerből kiindulva:

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

Az alapötlet, hogy itt most a ω_h rácsháló köztes értéket használjuk fel, jelen esetben a felezőpontokat, hiszen ez jobb közelítést tesz lehetővé, mintha a végpontok meredekségét alkalmaznánk.



Így a következő egyenlet áll elő

$$\frac{y_{n+0,5} - y_n}{0,5 \cdot h} = f(t_n, y_n), \quad (3.7.1)$$

amelyet rendezve $y_{n+0,5}$ -re a következőt kapjuk

$$y_{n+0,5} = y_n + 0,5h \cdot f(t_n, y_n). \quad (3.7.2)$$

Másrészt felhasználva a következő tulajdonságot

$$\frac{y_{n+1} - y_n}{h} = f(t_{n+0,5}, y_{n+0,5}), \quad (3.7.3)$$

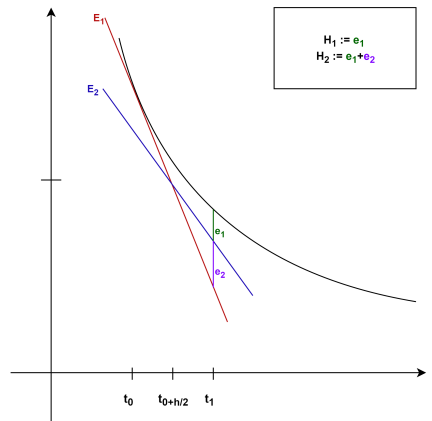
$$y_{n+1} = y_n + hf\left(t_{n+\frac{1}{2}}, y_{n+\frac{1}{2}}\right), \quad (3.7.4)$$

végül a 3.7.4 egyenletben $y_{n+\frac{1}{2}}$ helyére a 3.7.2 helyettesítést végrehajtva a következő eljárást kapjuk:

$$y_{n+1} = y_n + hf(t_n + 0,5h, y_n + 0,5hf(t_n, y_n)). \quad (3.7.5)$$

3.7.1. Definíció. (Javított Euler-módszer) A 3.7.5 iterációt javított Euler módszernek nevezzük.

A 7. ábrán a javított Euler-módszer geometriai interpretációja tekinthető meg. Lényegében olyan, mintha két explicit Euler-módszert alkalmaznánk, ami több műveletigényt eredményez, cserébe viszont nem kell megoldanunk nemlineáris egyenletet.



7. ábra. A javított Euler-módszer működésének vizualizációja

Vezessük be a 3.7.5 egyenlethez az alábbi jelöléseket és tároljuk az együtthatókat a következő táblázatban.

$$\begin{aligned}
k_1 &= f(t_n, y_n) \\
k_2 &= f(t_n + 0,5h, y_n + 0,5hk_1) \\
y_{n+1} &= y_n + h \cdot (0 \cdot k_1 + 1 \cdot k_2)
\end{aligned}$$

0	0	0
0,5	0,5	0
	0	1

A jobb oldalon lévő táblázatot Butcher-táblónak, vagy Butcher-táblának nevezzük. Az ilyen típusú módszereket általánosan a következőképp írjuk le.

3.7.2. Definíció. (*explicit s-lépcsős Runge-Kutta módszer*) Az alábbi formulát *explicit s-lépcsős Runge-Kutta módszernek* nevezzük:

$$\begin{aligned}
k_1 &= f(t_n, y_n) \\
k_2 &= f(t_n + a_2h, y_n + hb_{2,1}k_1) \\
k_3 &= f(t_n + a_3h, y_n + h(b_{3,1}k_1 + b_{3,2}k_2)) \\
&\vdots \\
k_s &= f(t_n + a_sh, y_n + h(b_{s,1}k_1 + b_{s,2}k_2 + \dots + b_{s,s-1}k_{s-1})) \\
y_{n+1} &= y_n + h(\sigma_1k_1 + \dots + \sigma_sk_s)
\end{aligned}$$

ahol $a_2, a_3, \dots, a_s, b_{2,1}, b_{3,1}, b_{3,2}, \dots, b_{s,1}, b_{s,2}, \dots, b_{s,s-1}$ tetszőleges, szabad paraméterek, σ_i ($i = 1, 2, \dots, n$)

0	0	0	0	0	0	0
a_2	$b_{2,1}$	0	0	0	0	0
a_3	$b_{3,1}$	$b_{3,2}$	0	0	0	0
a_4	$b_{4,1}$	$b_{4,2}$	$b_{4,3}$	0	0	0
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots
a_s	$b_{s,1}$	$b_{s,2}$	$b_{s,3}$	\dots	$b_{s,s-1}$	0
	σ_1	σ_2	σ_3	\dots	σ_s	0

2. táblázat. Általános explicit s-lépcsős Runge-Kutta módszer Butcher táblója

A javított Euler-módszerről következő mondható el konzisztencia szempontjából.

3.7.1. Állítás. A javított Euler-módszer másodrendben konzisztens.

Vannak-e még ilyen típusú másodrendű módszerek? A kérdést a következő állítás válaszolja meg.

3.7.2. Állítás. A 2-lépcsős, explicit Runge-Kutta módszer másodrendben konzisztens pontosan akkor, ha fennállnak az alábbi összefüggések:

1. $\sigma_1 + \sigma_2 = 1$
2. $\sigma_2 \cdot a_2 = 0,5$
3. $\sigma_2 \cdot b_{2,1} = 0,5$

3.7.3. Állítás. Az s-lépcsős, explicit Runge-Kutta módszer maximális pontossági rendje p:

- $1 \leq s \leq 4$ $p = s$
- $5 \leq s \leq 7$ $p = s - 1$
- $8 \leq s \leq 10$ $p = s - 2$

Az explicit Runge-Kutta típusú módszerek két másik nagyon fontos tulajdonsága a stabilitás és a konvergencia, amelyet csak vázlatosan ismertetek [13] alapján. A Runge-Kutta módszer stabilitási függvénye a következő

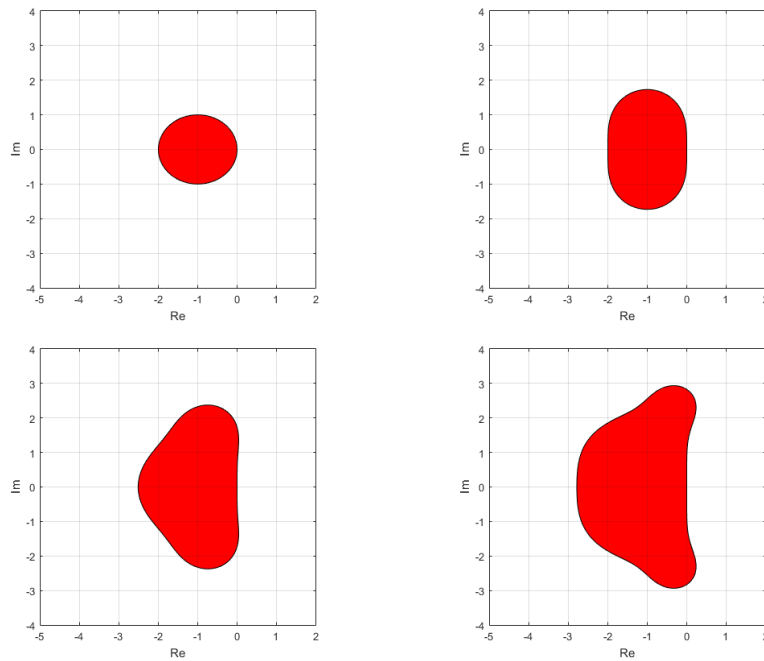
$$R(z) = 1 + \sum_{k=1}^p z^k \frac{1}{k!} + \sum_{k=p+1}^s z^k (b^T A^{k-1} e). \quad (3.7.6)$$

A 3.7.3 állítást felhasználva $p = s$ miatt $p = 1, 2, 3, 4$ esetén a második szumma eltűnik

$$R(z) = 1 + \sum_{k=1}^p z^k \frac{1}{k!}, \quad (3.7.7)$$

tehát az explicit Runge-Kutta típusú módszerek stabilitási függvénye egy legfeljebb s -edfokú polinom. Mivel $|z| \rightarrow \infty$ esetén $\lim |R(z)| = \infty$, ezért a stabilitási tartomány nem tartalmazza a teljes \mathbb{C}^- félsíkot, így a következő mondható el.

3.7.4. Állítás. *Az explicit Runge-Kutta típusú módszerek nem A-stabilak.*



8. ábra. Az első-, másod-, harmad-, és negyedrendű explicit Runge-Kutta módszerek abszolút stabilitási tartománya.

Ahogy a 8. ábrán is látható a rend növelésével együtt nő a stabilitási tartomány mérete, azonban a stabilitási tartománynak nem lesz része a teljes baloldali félsík.

Az explicit Runge-Kutta típusú módszerek konvergenciájáról az alábbi mondható el.

3.7.1. Tétel. *Tegyük fel, hogy a kezdetiérték-feladatban f folytonos függvény és a második változójában Lipschitzes. Ekkor egy p -ed rendben konzisztens explicit Runge-Kutta típusú módszer p -ed rendben konvergens is.*

3.8. Lineáris többlépéses módszerek

Eddig olyan numerikus módszerekről volt szó, amelyeknél az aktuális rácsháló-pontbeli közelítéshez (t_{n+1}) csak az öt közvetlen megelőző rácspontot (t_n) használtuk fel. Noha a Runge-Kutta típusú módszereknél a két rácspont között s darab pontban értékeket számítottunk ki és azokat is felhasználtuk, de a fő különbség, hogy itt csak w_h rácsháló pontokat alkalmazunk. A lineáris többlépéses módszerekről a legfontosabb ismereteket [12] és [13] alapján foglalom össze.

3.8.1. Definíció. (Lineáris többlépéses módszer) A következő képlettel értelmezett eljárást lineáris többlépéses módszernek nevezzük

$$\frac{a_0 y_n + a_1 y_{n-1} + \dots + a_m y_{n-m}}{h} = b_0 f_n + b_1 f_{n-1} + \dots + b_m f_{n-m}, \quad (3.8.1)$$

ahol $n = m, m+1, \dots$ és $a_0, a_1, \dots, a_m, b_0, b_1, \dots, b_m$ adott rögzített paraméterek.

Abban az esetben, ha $b_0 = 0$ explicit lineáris többlépéses módszerről beszélünk, $b_0 \neq 0$ esetén a módszer implicit.

3.8.1. Példa. $m = 1$ esetén nyilvánvalóan egylépéses módszereket kapjuk vissza, ahol a 3.8.1 a következőképp áll elő

$$\frac{a_0 y_n + a_1 y_{n-1}}{h} = b_0 f_n + b_1 f_{n-1}.$$

Az alábbi megválasztásokkal, az eddig ismertett egylépéses módszereket kapjuk vissza:

$a_0 = 1, a_1 = -1, b_0 = 0$ és $b_1 = 1$ - explicit Euler-módszer,

$a_0 = 1, a_1 = -1, b_0 = 1$ és $b_1 = 0$ - implicit Euler-módszer,

$a_0 = 1, a_1 = -1, b_0 = 0,5$ és $b_1 = 0,5$ - Crank-Nicolson-módszer.

Tekintsük a paraméterek első megválasztás és legyen $a_0 = 2, a_1 = -2, b_0 = 0$ és $b_1 = 1$ paramétereket alkalmazva

$$\frac{2y_n - 2y_{n-1}}{h} = 2f_{n-1},$$

amely szintén az explicit Euler-módszert adja és 2-vel leosztva

$$\frac{y_n - y_{n-1}}{h} = f_{n-1}.$$

Tehát a két eljárást ugyanazt az eredményt adja, ezért azt mondjuk, hogy két numerikus módszer ekvivalens, ha a kezdeti adatokból ugyanazt a megoldást adják, azaz megválasztások konstansszorosa között nem teszünk különbséget. ♦

A 3.8.1 példában bemutatott problémát elkerülendő, a $2(m+1)$ paraméterből az egyiket lefixáljuk az egyértelműséget biztosítva az $a_0 = 1$ normalizáló feltétellel, ezáltal már csak $2m+1$ szabad paraméter áll a rendelkezésünkre. Szokás még az $a_0 = 1$ helyett egy másik normalizáló feltételt megadni,

$$\sum_{k=0}^m b_k = 1. \quad (3.8.2)$$

A módszer ismertetése után jöjjön annak három legfontosabb tulajdonsága; konzisztencia, stabilitás és konvergencia.

3.8.1. Tétel. (A p -ed rendűség feltétele) Azt mondjuk, hogy a 3.8.1 lineáris többlépéses numerikus módszer p -ed rendben konzisztens, ha a paraméterekre teljesülnek az alábbi feltételek:

$$a_0 = 1 \quad (3.8.3)$$

$$\sum_{k=0}^m a_k = 0 \quad (3.8.4)$$

$$\frac{1}{j} \sum_{k=0}^m k^j a_k + \sum_{k=0}^m k^{j-1} b_k = 0 \quad j = 1, 2, \dots, p. \quad (3.8.5)$$

3.8.2. Tétel. Egy m -lépéses implicit lineáris többlépéses módszer maximális rendje $2m$, az explicit lineáris többlépéses módszeré $2m - 1$.

Vezessük be a következő kép polinomot

$$\varrho(\xi) = \sum_{k=0}^m a_k \xi^{m-k}, \quad (3.8.6)$$

$$\sigma(\xi) = \sum_{k=0}^m b_k \xi^{m-k}. \quad (3.8.7)$$

3.8.2. Definíció. (Első és második karakterisztikus polinom) 3.8.6 és 3.8.7 legfeljebb m -edfokú polinomokat az m -lépéses lineáris többlépéses módszer első illetve második karakterisztikus polinomjának nevezzük.

3.8.3. Tétel. Egy lineáris többlépéses módszer pontosan akkor konzisztens, amikor a karakterisztikus polinomjaira érvényesek az alábbi összefüggések

$$\varrho(1) = 0 \quad \text{és} \quad \varrho'(1) = \sigma(1) \quad (3.8.8)$$

3.8.3. Definíció. Azt mondjuk, hogy egy lineáris többlépéses módszer kielégíti a gyökkritériumot, ha a $\varrho(\xi) = 0$ karakterisztikus egyenlet $\xi_k \in \mathbb{C}$ gyökeire a $|\xi_k| \leq 1$ és a $|\xi_k| = 1$ tulajdonságú gyökök egyszeresek.

3.8.4. Tétel. Ha egy lineáris többlépéses módszer konzisztens és érvényes rá a gyökkritérium, akkor konvergens is.

3.8.4. Definíció. (Erősen stabil módszer) Azt mondjuk, hogy egy lineáris többlépéses módszer erősen stabil, ha kielégíti a gyökkritériumot és csak a $\xi = 1$ az egyetlen egy abszolút értékű gyöke.

3.8.5. Tétel. Egy erősen stabil m -lépéses lineáris többlépéses módszer legfeljebb $m + 1$ -ed rendű lehet.

3.8.6. Tétel. Az explicit lineáris többlépéses módszerek nem A -stabilak, illetve az A -stabil lineáris többlépéses módszerek maximális rendje 2.

Vegyük a 3.8.1 lineáris többlépéses módszert az alábbi megválasztásokkal

$$a_0 = 1, \quad a_1 = -1, \quad a_2 = \dots = a_n = 0. \quad (3.8.9)$$

3.8.5. Definíció. (Adams-típusú módszerek) A 3.8.1 iterációt a 3.8.9 megválasztásokkal Adams-típusú módszereknek nevezzük:

$$\frac{y_n - y_{n-1}}{h} = \sum_{k=0}^m b_k f_{n-k}. \quad (3.8.10)$$

Adams-típusú módszerek esetén $m + 1$ megválasztható paraméter van, a b_k együtthatók. Itt is nyilván $b_0 = 0$ esetén explicit (Adams-Bashforth-módszer), $b_0 \neq 0$ esetén implicit a módszer (Adams-Moulton-módszer).

4. Számítógépes modell: Cottrell-kísérlet

A szimuláció alapjául [1] szolgált, kiegészítve [15] parciális differenciálegyenletek ismertetésével.

4.1. Az analitikus megoldás

A Cottrell-kísérlet az egyik legrégebbi elektrokémiai probléma, amit legelőször Frederick Gardner Cottrell ismertetett 1903-ban. Először a kísérlet egy ismert analitikus megoldását fogom levezetni, majd ezután a szimulációt elvégezve az eredmények közvetlenül összehasonlíthatóak az analitikus megoldással. Fick II. törvényét az alábbi másodrendű parciális differenciálegyenlet írja le:

$$\frac{\partial c(x, t)}{\partial t} = D \cdot \frac{\partial^2 c(x, t)}{\partial x^2}, \quad (4.1.1)$$

ahol x térbeli és t időbeli változó, $c(x, t)$ függvény a koncentrációt jelöli, valamint D a diffúziós együttható.

Megjegyzés. Az analitikus megoldás felvázolása előtt tekintsük át az általános diffúziós egyenletet. Legyen t az időbeli változó, x_1, \dots, x_n az n darab térbeli változó, c a koncentráció. Az áttekinthetőség érdekében legyen $c := c(x_1, \dots, x_n, t)$:

$$\frac{\partial c}{\partial t} = \sum_{i=1}^n \frac{\partial}{\partial x_i} \left(D(c, x_1, \dots, x_n, t) \frac{\partial c}{\partial x_i} \right)$$

Szakedolgozatomban azt az esetet fogom vizsgálni, amikor az időbeli változó mellett csak egy darab térbeli változó van, illetve a D diffúziós együttható nem függ semmitől, egy konstans, valós szám.

A 4.1.1 egyenlet a diffúziós egyenlet, amely megadja a koncentráció időbeli változását egy adott helyen az ott lévő hely szerinti koncentráció változás függvényében, tehát a koncentrációváltozás sebessége arányos az adott hely szerinti második deriváltjával.

Megjegyzés. Tekintsük a lineáris másodrendű parciális differenciálegyenletek általános alakját

$$\sum_{j,k=1}^n a_{jk} \partial_j \partial_k u + \sum_{j=1}^n b_j \partial_j u + cu = f,$$

ahol a_{jk}, b_j, c és $f : \Omega \rightarrow \mathbb{R}$ adott függvények. A másodrendű főrészükben lineáris és másodrendű lineáris parciális differenciálegyenleteket a főrészük alapján osztályozzuk. Legyen x_0 rögzített és vezessük be az $A(x_0)$ jelölést az előbb felírt parciális differenciálegyenlet általános alakjában a főrész együtthatóiból képzett $n \times n$ mátrixra

$$A(x_0) := [a_{jk}(x_0)]_{j,k=1}^n.$$

Továbbá vezessük be az $n^+(x), n^-(x)$ és $n^0(x)$ jelöléseket rendre a pozitív, negatív és 0 sajátértékek számára. Az egyenleteket a következőképp osztályozzuk

- 1) elliptikus, ha $n^+ = n$ vagy $n^- = n$
- 2) hiperbolikus, ha $(n^+ = n - 1$ és $n^- = 1)$ vagy $(n^+ = 1$ és $n^- = n - 1)$
- 3) parabolikus, ha $n^0 = 1$ és $(n^+ = n - 1$ vagy $n^- = n - 1)$

A diffúziós egyenlet egy parabolikus egyenlet.

A kitérők után oldjuk meg analitikusan a 4.1.1. parciális differenciálegyenletet. Felírva a koncentráció t -szerinti Laplace-transzformációját az alábbi összefüggést kapjuk

$$C(x, s) = \int_0^{\infty} c(x, t) \cdot e^{-st} dt.$$

A bal oldalra alkalmaztuk a generátorfüggvény első deriváltjára vonatkozó összefüggést, jobb oldalon a koncentráció Laplace-transzformáltját:

$$s \cdot C(x, s) - \underbrace{C(x, 0)}_{c_{\infty}} = D \cdot \frac{\partial^2 C(x, s)}{\partial x^2},$$

rendezve a differenciálegyenletet:

$$\frac{\partial C(x, s)}{\partial x^2} - \frac{s}{D} \cdot C(x, s) = -\frac{c_{\infty}}{D}. \quad (4.1.2)$$

A Laplace-transzformáció alkalmazásának köszönhetően a kiindulási másodrendű parciális differenciálegyenletet visszavezettük egy könnyebb feladat megoldására, hiszen a 4.1.2 egy egyváltozós másodrendű közönséges differenciálegyenlet. Az inhomogén lineáris másodrendű differenciálegyenlet megoldása előáll a homogén egyenlet általános megoldása és az inhomogén egyenlet partikuláris megoldásaként, általánosan:

$$\varphi(t) = \underbrace{\varphi_0(t)}_{\text{inhomogén egyenlet partikuláris megoldása}} + \underbrace{A \cdot \varphi_1(t) + B \cdot \varphi_2(t)}_{\text{homogén egyenlet általános megoldása}}.$$

A homogén egyenlet:

$$\frac{\partial C(x, s)}{\partial x^2} + 0 \cdot \frac{\partial C(x, s)}{\partial x} - \frac{s}{D} \cdot C(x, s) = 0, \quad (4.1.3)$$

Keressük meg a 4.1.3 egyenlet $C(x, s) = e^{\lambda x}$ alakú megoldásait, vagyis alkalmazva ezt a behelyettesítést:

$$\lambda^2 \cdot e^{\lambda x} + 0 \cdot \lambda e^{\lambda x} - \frac{s}{D} \cdot e^{\lambda x} = 0$$

Osszunk le $e^{\lambda x}$ -szel - amit ugye nyugodtan megtehetünk, mivel $e^{\lambda x} > 0$ - a következő másodfokú egyenletet kapjuk, amely a differenciálegyenlet karakterisztikus egyenlete:

$$\lambda^2 + 0 \cdot \lambda - \frac{s}{D} = 0,$$

melynek gyökei: $\lambda_1 = \sqrt{\frac{s}{D}}$ és $\lambda_2 = -\sqrt{\frac{s}{D}}$, így a lineáris másodrendű differenciálegyenlet lineárisan független alaprendszer: $\varphi_1 = \exp\left(x \cdot \sqrt{\frac{s}{D}}\right)$ és $\varphi_2 = \exp\left(-x \cdot \sqrt{\frac{s}{D}}\right)$, ezáltal a homogén egyenlet általános megoldása:

$$A \cdot \exp\left(x \cdot \sqrt{\frac{s}{D}}\right) + B \cdot \exp\left(-x \cdot \sqrt{\frac{s}{D}}\right).$$

Az inhomogén egyenlet partikuláris megoldása $\frac{c_{\infty}}{s}$, így a 4.1.2 differenciálegyenlet általános megoldása:

$$C(x, s) = \frac{c_{\infty}}{s} + A \cdot \exp\left(x \cdot \sqrt{\frac{s}{D}}\right) + B \cdot \exp\left(-x \cdot \sqrt{\frac{s}{D}}\right), \quad (4.1.4)$$

ahol A és B egyelőre ismeretlen valós számok. Ez a két szám az alábbi két peremfeltételből egyszerűen meghatározható:

- az elektródtól távoli tartományra nincs hatással az elektródfolyamat, azaz a koncentráció értéke $x \rightarrow \infty$ esetén c_∞ ,
- az elektród felületen a koncentráció 0 bármely $t > 0$ esetén,

vagyis pontosabban:

$$t > 0 \text{ esetén : } \lim_{x \rightarrow \infty} c(x, t) = c_\infty \rightarrow \lim_{x \rightarrow \infty} C(x, s) = \frac{c_\infty}{s} \quad (4.1.5)$$

$$t > 0 \text{ esetén : } \lim_{x \rightarrow 0} c(x, t) = 0 \rightarrow \lim_{x \rightarrow 0} C(x, s) = 0 \quad (4.1.6)$$

4.1.5 és 4.1.6 feltételekből meghatározhatjuk az eddig ismeretlen A és B együtthatók értékét, amelyek a következők: $A = 0$ és $B = -\frac{c_\infty}{s}$. Ezeket behelyettesítve a 4.1.4 általános megoldásba:

$$C(x, s) = \frac{c_\infty}{s} - \frac{c_\infty}{s} \cdot \exp\left(-x \cdot \sqrt{\frac{s}{D}}\right) = \frac{c_\infty}{s} \left(1 - \exp\left(-x \cdot \sqrt{\frac{s}{D}}\right)\right). \quad (4.1.7)$$

Ezek után transzformáljuk vissza a 4.1.7 egyenletet, tagonként véve az inverz Laplace transzformáltat, megkapjuk $c(x, t)$ függvényt. A bal oldal inverz transzformáltja triviális, a jobboldalon pedig az így keresett

$$L^{-1} \left[\frac{c_\infty}{s} \right] = c_\infty \cdot L^{-1} \left[\frac{1}{s} \right] = c_\infty \cdot 1 = c_\infty,$$

$$L^{-1} \left[\frac{c_\infty}{s} \cdot \exp\left(-x \sqrt{\frac{s}{D}}\right) \right] = c_\infty \cdot \operatorname{erfc}\left(\frac{x}{2\sqrt{Dt}}\right),$$

inverz transzformációkat végrehajtva megkapjuk a keresett $c(x, t)$ függvényt:

$$c(x, t) = c_\infty \left[1 - \operatorname{erfc}\left(\frac{x}{2\sqrt{Dt}}\right) \right]. \quad (4.1.8)$$

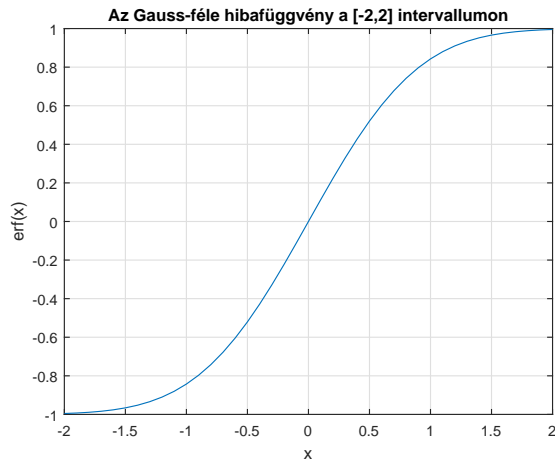
A 4.1.8 egyenletben szereplő $\operatorname{erfc}(x)$ a komplementer hibafüggvény és felhasználva az $\operatorname{erfc}(x) = 1 - \operatorname{erf}(x)$ összefüggést a 4.1.8 a következőképp áll elő:

$$c(x, t) = c_\infty \operatorname{erf}\left(\frac{x}{2\sqrt{Dt}}\right). \quad (4.1.9)$$

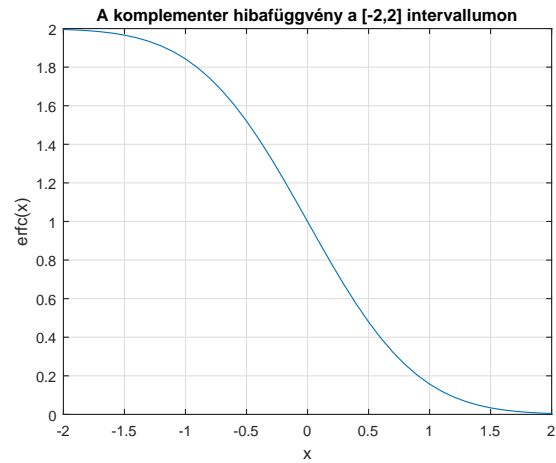
A 4.1.9. egyenletben lévő $\operatorname{erf}(x)$ függvény a Gauss-féle hibafüggvény, amely az alábbi integrállal definiálható:

$$\operatorname{erf}(z) = \frac{2}{\sqrt{\pi}} \int_0^z e^{-u^2} du. \quad (4.1.10)$$

Az előbb definiált hibafüggvények grafikonja szerepel a 9. és a 10. ábrákon.

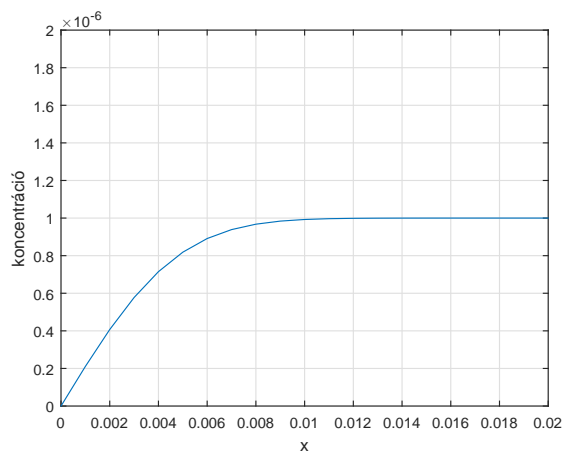


9. ábra. A Gauss-féle hibafüggvény $\text{erf}(x)$

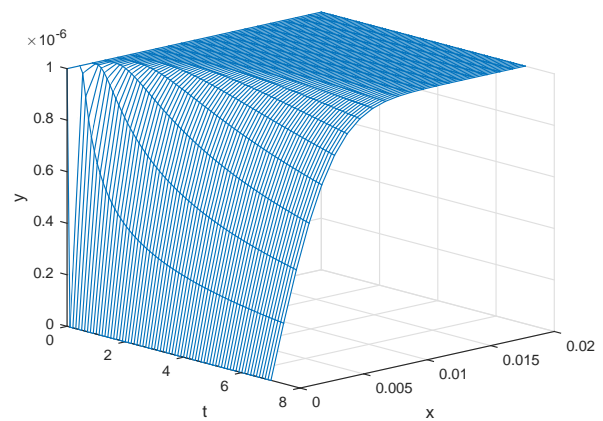


10. ábra. A komplementer hibafüggvény $\text{erfc}(x)$

A 11-12. ábrákon látható a koncentráció függvény, illetve háromdimenziós grafikonja.



11. ábra. A koncentráció függvény grafikonja a rögzített $T=7$ időpillanatban $c(x, 7)$.



12. ábra. A koncentráció függvény grafikonja három dimenzióban

4.2. A kísérlet szimulációja

4.2.1. A numerikus megoldás alapja

Vizsgálódásunk tárgyát továbbra is a diffúziós egyenlet, vagyis a

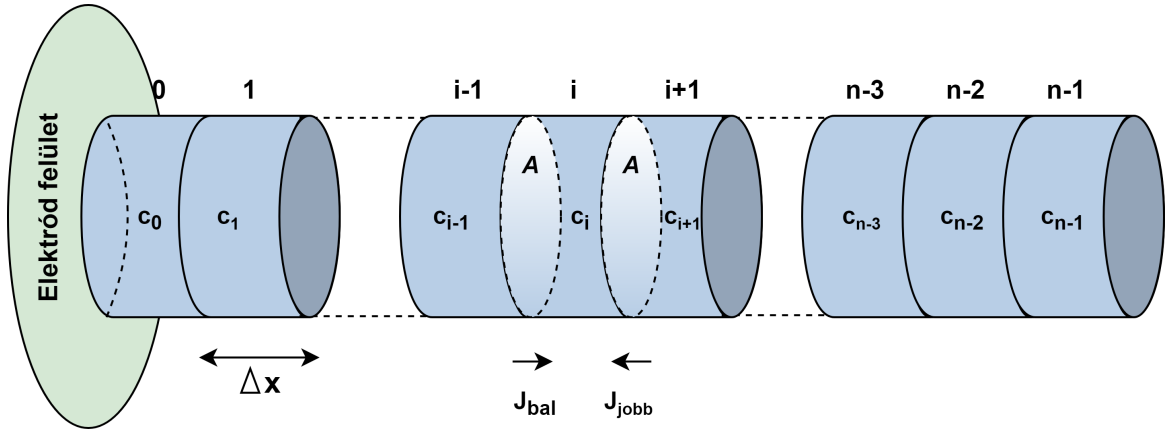
$$\frac{\partial c(x, t)}{\partial t} = D \cdot \frac{\partial^2 c(x, t)}{\partial x^2}$$

parciális differenciálegyenlet képzí. Az előző alfejezetben előállítottuk az analitikus megoldást, a következőkben a numerikus megoldások ismertetésére kerül sor. Ezeket a közelítéseket vetjük össze egyfelől a pontos megoldással, illetve ezek egymáshoz viszonyított teljesítményét és viselkedését elemezzük.

Legyen $c \in \mathbb{R}^{n+1}$ a koncentrációvektor, amelyben kezdetben az első elem kivételével minden komponens értéke a tömeg koncentrációval, c_∞ -tel egyenlő. Felhasználva az egyik peremfeltételt, miszerint a koncentráció az elektród felületén nulla minden $t > 0$ időpillanatban, így a következő kezdővektort kapjuk a koncentrációra:

$$c^{(0)} = (0, c_\infty, c_\infty, \dots, c_\infty)^T. \quad (4.2.1)$$

Ne feledjük, hogy nem csak a kezdővektorban 0 az első elem, hanem minden egyes iteráció elvégeztével is a koncentrációvektorban.



13. ábra. A Cottrell-kísérlet digitális szimulációjának felvázolása

A 13. ábra szemlélteti a kísérlet szimulációját. Ismerjük kezdetben a koncentrációt $n + 1$ helyen, azt fogjuk megvizsgálni, hogy adott helyen hogyan változik a koncentráció Δt időlépések elteltével. Felhasználva Fick I. törvényét, ami az anyagfluxus (anyagáramlás) és a koncentráció gradiens közti arányosságot írja le, a következő összefüggéseket kapjuk

$$J_{\text{bal}} = \frac{\Delta n_{\text{bal}}}{A \cdot \Delta t} = -D \cdot \frac{c_i - c_{i-1}}{\Delta x}, \quad (4.2.2)$$

$$J_{\text{jobb}} = \frac{\Delta n_{\text{jobb}}}{A \cdot \Delta t} = -D \cdot \frac{c_i - c_{i+1}}{\Delta x}. \quad (4.2.3)$$

4.2.2 a i helyre beáramló anyagáramlást írja le a $i - 1$ helyről, vagyis a bal szomszédjából, míg a 4.2.3 a i helyről a $i + 1$ pontba, vagyis a jobb szomszédba távozó fluxust. A Δn_{bal} és Δn_{jobb} az anyagmennyiséget jelölik. Az i -edik helyen minden időpillanatban $\Delta n = \Delta n_{\text{bal}} + \Delta n_{\text{jobb}}$ anyagmennyiség áramlik.

$$\frac{\Delta c_i}{\Delta t} = \frac{(J_{\text{bal}} + J_{\text{jobb}}) \cdot A}{A \Delta x} = D \cdot \frac{c_{i-1} - 2c_i + c_{i+1}}{\Delta x^2} \quad (4.2.4)$$

4.2.4 azt írja le, hogy az egyes időlépésekben az i -edik helybeli koncentrációváltozás függ az i -edik, $i - 1$ -edik és $i + 1$ -edik helybeli koncentrációktól, továbbá a D , Δx és Δt paraméterektől. Az 4.2.4. egyenlet diszkretizált változata a diffúziós egyenletnek. Az eddigi levezetés matematikailag csupán annyit jelent, hogy a 4.1.1. diffúziós egyenlet jobb oldalát diszkretizáltuk, azaz a másodrendű térbeli deriváltat a másodrendű központi differenciával közelítettük. A könnyebb átláthatóság kedvéért írjuk át a 4.2.4. egyenletet mátrix-vektor formába

$$\frac{\Delta c}{\Delta t} = \frac{D}{\Delta x^2} \cdot L \cdot c, \quad (4.2.5)$$

ahol c a koncentrációvektor és L az úgynevezett Laplace mátrix. A Laplace mátrix a következőképp definiálható

$$L_{i,j} = \begin{cases} 1 & \text{ha } i = j \pm 1 \\ -\deg(i) & \text{ha } i = j \\ 0 & \text{egyébként,} \end{cases} \quad (4.2.6)$$

ahol $\deg(i)$ az i -edik helyen lévő szomszédok száma. Az L Laplace mátrix a következőképp néz ki:

$$L = \begin{pmatrix} -1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & -2 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -2 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & \ddots & \ddots & 0 & 0 \\ 0 & 0 & 0 & \ddots & \ddots & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & -2 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{pmatrix}. \quad (4.2.7)$$

Az 4.2.5 egyenlet jobboldala a koncentráció második térbeli deriváltját approximálja, tehát a 4.1.1 diszkretizált verziója, ami Fick második törvényét írja le, a bal oldal, pedig az idő szerinti derivált véges differenciás approximációja. Elenyészően kicsi lépésközöket alkalmazva a következő összefüggést kapjuk:

$$\frac{\partial c}{\partial t} = \frac{D}{\Delta x^2} \cdot L \cdot c, \quad (4.2.8)$$

ami egy differenciálegyenlet a c vektorra nézve. Tegyük fel, hogy a k -adik lépésben ismerjük a $c^{(k)}$ koncentrációvektort. A következő lépésben Δt időegységgel később, a $c^{(k+1)}$ vektor meghatározható a 4.2.8 differenciálegyenletből a következő iterációval:

$$c^{(k+1)} = \exp(L_{mod}) \cdot c^{(k)}, \quad (4.2.9)$$

ahol L_{mod} a súlyozott Laplace mátrix:

$$L_{mod} = \frac{D\Delta t}{\Delta x^2} \cdot L. \quad (4.2.10)$$

Vezessük be a következő jelölést

$$S := \exp(L_{mod}),$$

ahol S -t 4.2.9. egyenlethez tartozó lépésmátrixnak nevezzük.

A következő alfejezetben ismertetem a szimuláció paramétereit, azt követi a numerikus módszerek bemutatása, végezetül ezek teljesítményét hasonlítom össze egymással, illetve az analitikus megoldással.

4.2.2. A szimuláció paraméterei

Ebben az alfejezetben fogom ismertetni a kísérlet szimulációjának alapját és paramétereit. A szimuláció standard bemeneti paramétereit a 3. táblázat tartalmazza. Az egyes esetek összehasonlítása során egy-két paraméter értékét megváltoztatjuk (például nevezetesen ilyenek a lépésközök vagy esetleg a végpontok), ezek minden esetben feltüntetésre kerülnek, a többi paraméter értéke a 3. táblázat adatain alapul.

Paraméter	Jele	Értéke
Térbeli lépésköz	Δx	0,001
Térbeli lépésközök száma	n	100
Térbeli kezdőpont	X_0	0
Térbeli végpont	X_n	0,1
Diffúziós együttható	D	10^{-6} cm ² /mp
Tömeg koncentráció	c_∞	10^{-6} mol/cm ³
Időbeli lépésköz	Δt	0,1 mp
Időbeli lépésközök száma	T	70
Időbeli kezdőpont	T_0	0 mp
Időbeli végpont	T_n	7 mp

3. táblázat. A szimuláció bemenő paraméterei

Vezessük be a $(Tmtx, Xmtx, Ymtx)$ mátrixhármast, ahol $Tmtx \in \mathbb{R}^{101 \times 71}$ az időbeli változó mátrix, $Xmtx \in \mathbb{R}^{101 \times 71}$ a térbeli változó mátrix és $Ymtx \in \mathbb{R}^{101 \times 71}$ a koncentrációvektorokat tartalmazó mátrix, amelyek a következőképp állnak elő:

$$Tmtx = \begin{pmatrix} 0 & 0,1 & 0,2 & \dots & 6,8 & 6,9 & 7 \\ 0 & 0,1 & 0,2 & \dots & 6,8 & 6,9 & 7 \\ 0 & 0,1 & 0,2 & \dots & 6,8 & 6,9 & 7 \\ \vdots & \vdots & \vdots & \dots & \vdots & \vdots & \vdots \\ 0 & 0,1 & 0,2 & \dots & 6,8 & 6,9 & 7 \\ 0 & 0,1 & 0,2 & \dots & 6,8 & 6,9 & 7 \\ 0 & 0,1 & 0,2 & \dots & 6,8 & 6,9 & 7 \end{pmatrix}, \quad Xmtx = \begin{pmatrix} 0 & 0 & \dots & 0 & 0 & 0 \\ 0,001 & 0,001 & \dots & 0,001 & 0,001 & 0,001 \\ 0,002 & 0,001 & \dots & 0,002 & 0,002 & 0,002 \\ \vdots & \vdots & \dots & \vdots & \vdots & \vdots \\ 0,098 & 0,098 & \dots & 0,098 & 0,098 & 0,098 \\ 0,099 & 0,099 & \dots & 0,099 & 0,099 & 0,099 \\ 0,1 & 0,1 & \dots & 0,1 & 0,1 & 0,1 \end{pmatrix},$$

$$Ymtx = \begin{pmatrix} c_0^{(0)} & c_0^{(1)} & c_0^{(2)} & \dots & c_0^{(68)} & c_0^{(69)} & c_0^{(70)} \\ c_1^{(0)} & c_1^{(1)} & c_1^{(2)} & \dots & c_1^{(68)} & c_1^{(69)} & c_1^{(70)} \\ c_2^{(0)} & c_2^{(1)} & c_2^{(2)} & \dots & c_2^{(68)} & c_2^{(69)} & c_2^{(70)} \\ \vdots & \vdots & \vdots & \dots & \vdots & \vdots & \vdots \\ c_{98}^{(0)} & c_{98}^{(1)} & c_{98}^{(2)} & \dots & c_{98}^{(68)} & c_{98}^{(69)} & c_{98}^{(70)} \\ c_{99}^{(0)} & c_{99}^{(1)} & c_{99}^{(2)} & \dots & c_{99}^{(68)} & c_{99}^{(69)} & c_{99}^{(70)} \\ c_{100}^{(0)} & c_{100}^{(1)} & c_{100}^{(2)} & \dots & c_{100}^{(68)} & c_{100}^{(69)} & c_{100}^{(70)} \end{pmatrix}.$$

Ebből a három koordinátamátrixból vázolható fel a háromdimenziós ábra. Nyilvánvalóan a bemenő paraméterek megváltozásával a mátrixok is módosulni fognak. A fenti három mátrix a 3. táblázat adatai alapján áll elő, pusztán szemléltetésre szolgálnak. A fejezet további alfejezeteiben az eddig bemutatott numerikus módszereket fogjuk alkalmazni, illetve kielemezni.

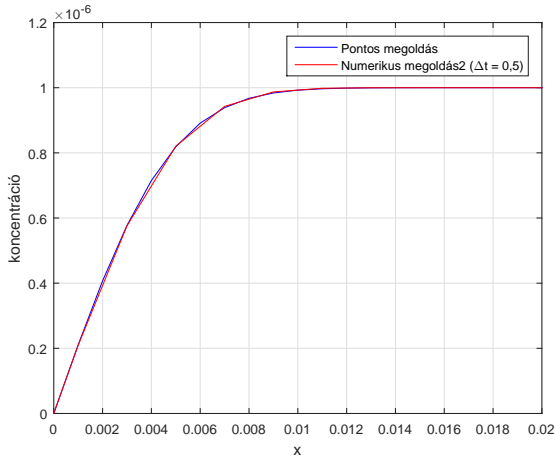
4.2.3. Az explicit Euler-módszer

Az explicit Euler-módszer iterációját a 4.2.11. egyenlettel írhatjuk le

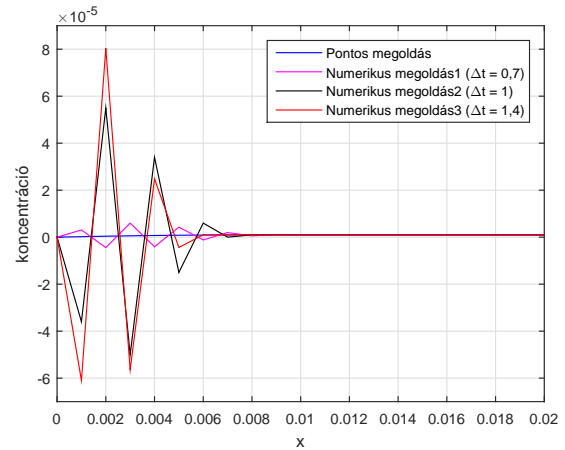
$$c^{(k+1)} = c^{(k)} + L_{mod} \cdot c^{(k)} = (I + L_{mod}) \cdot c^{(k)}. \quad (4.2.11)$$

A módszer egyik nagy előnye, hogy a következő időbeli - jelen esetben pont helyett vektor - közelítés egy egyszerű behelyettesítéssel kiszámolható. Azonban ennek az eljárásnak az alkalmazása

az egyik leghátrányosabb, ugyanis $\Delta t > \frac{\Delta x^2}{2D}$ esetén az explicit Euler-módszer nem lesz konvergens - ami a 3. táblázatban szereplő paraméterek alapján $\Delta t > 0,5$ - és erősen oszcillálva közelíti a pontos megoldást, ahogy az a 14-15. ábrán is látható.

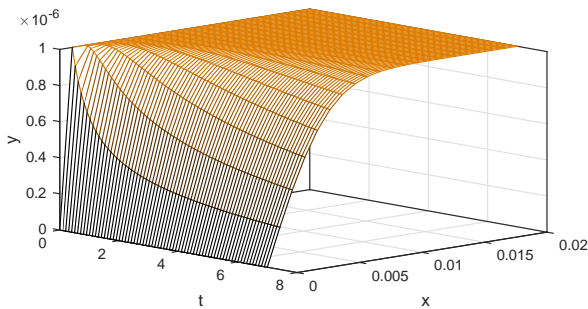


14. ábra. Az explicit Euler-módszerrel történő közelítés $\Delta t = 0,5$ lépésközt alkalmazva a $T=7$ időpillanatban.

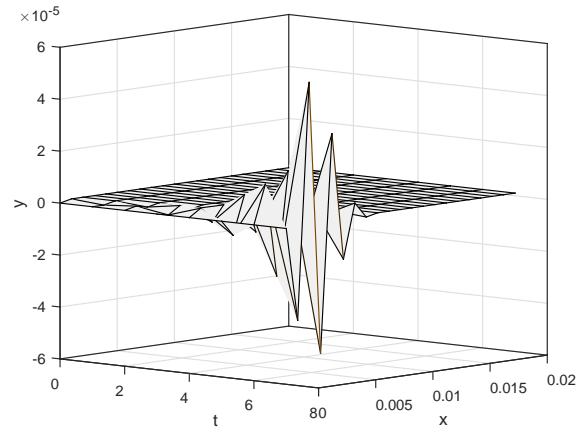


15. ábra. Az explicit Euler-módszerrel történő közelítés $\Delta t = 0,7, 1, 1,4$ lépésközt alkalmazva a $T=7$ időpillanatban.

Ahogy azt a 15. ábra is szemlélteti, $\Delta t = 0,5$ lépésközt esetén a numerikus módszer jól közelíti a pontos megoldást, azonban ennél nagyobb Δt esetén - már 0,2-nél nagyobb lépésközt esetén is - hatalmas ingadozások fedezhetők fel az approximáció során. Ez a rendszertelen viselkedés annak tudható be, hogy az explicit Euler-módszer nem A -stabil eljárás, ezáltal az input paraméterek megváltozásával nem változik arányosan az output értéke.



16. ábra. Az explicit Euler-módszerrel történő közelítés $\Delta t = 0,1$ esetén.



17. ábra. Az explicit Euler-módszerrel történő közelítés $\Delta t = 1$ esetén.

A 16-17. ábrákon megfigyelhető általánosan a közelítés. Amíg az első esetben egy viszonylag kis időlépésközt alkalmaztunk, ahogy az várható, az eljárás jól közelíti a pontos megoldást. A második esetben már nagyobb lépésköz esetén a módszer kaotikusan viselkedik és magas kiugró értékeket produkál.

4.2.4. Az implicit Euler-módszer

Az implicit Euler-módszerrel történő iteráció a következőképp néz ki

$$c^{(k+1)} = c^{(k)} + L_{mod} \cdot c^{(k+1)}.$$

A könnyebb implementálhatóság végett célszerű átvinni a jobboldalon lévő második tagot a bal oldalra

$$(I - L_{mod}) \cdot c^{(k+1)} = c^{(k)},$$

majd ezt kifejezve a $(k+1)$ -edik vektorra

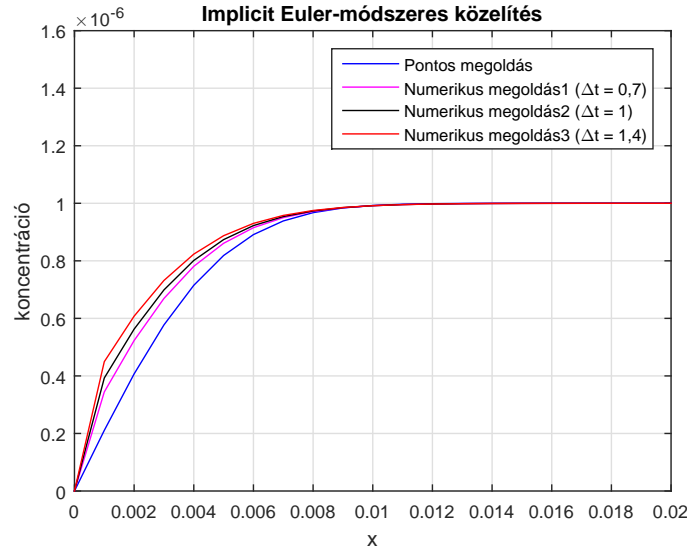
$$c^{(k+1)} = (I - L_{mod})^{-1} \cdot c^{(k)}, \quad (4.2.12)$$

ahol

$$S := (I - L_{mod})^{-1}$$

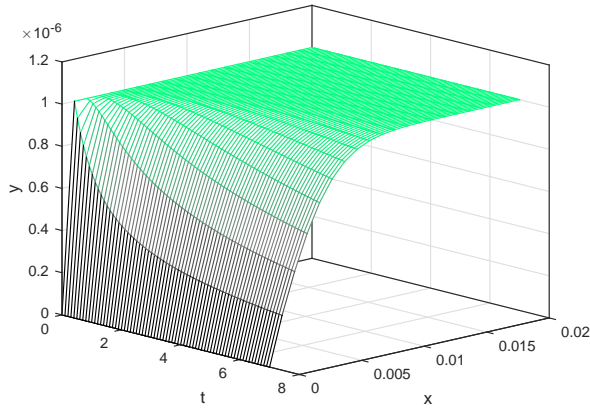
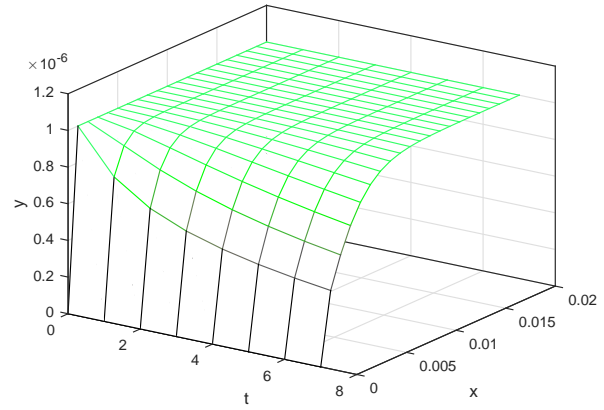
a 4.2.12. egyenlethez tartozó lépésmátrix.

A 18. ábra jól szemlélteti az explicit és az implicit Euler-módszer közötti különbséget. Amíg az explicit Euler módszer esetében 0,5-nél nagyobb lépésköz esetén a módszer nem közelíti a megoldást és oszcillálni kezd, az implicit Euler-módszert illetően a lépésközők megválasztásával arányosan változik a közelítés. Ez a tulajdonság az implicit módszer A -stabilitásának köszönhető, az explicit módszer nem A -stabil, ezért nem változott a bemenő paraméterek változtatásával arányosan a közelítés.



18. ábra. Az implicit Euler-módszerrel történő közelítés $\Delta t = 0,7$, 1 , és $1,4$ lépésközőkkel a $T=7$ időpillanatban.

A háromdimenziós ábrákon is tapasztalható ez a különbség, hiszen még $\Delta t = 1$ esetben sem torzul el a görbe, mint ahogy az az explicit Euler-módszer alkalmazása esetén történt.

19. ábra. Az implicit Euler-módszerrel történő közelítés $\Delta t = 0,1$ esetén.20. ábra. Az implicit Euler-módszerrel történő közelítés $\Delta t = 1$ esetén.

Ugyan mindkét eljárás elsőrendű, a 4. táblázat adatait látva elmondható, hogy jelen esetben kisebb lépésközök esetén az explicit Euler-módszer pontosabb, 0,5-nél nagyobb lépésköz esetén teljesít jobban az implicit módszer.

Pontos	explicit Euler-módszer					implicit Euler-módszer				
	h=0,1	h=0,2	h=0,5	h=0,7	h=1	h=0,1	h=0,2	h=0,5	h=0,7	h=1
0	0	0	0	0	0	0	0	0	0	0
0,2107	0,2102	0,2092	0,2095	3,0745	-36,00	0,2319	0,2520	0,3091	0,3444	0,3928
0,4070	0,4060	0,4042	0,3928	-4,4382	55,00	0,4263	0,4441	0,4935	0,5230	0,5626
0,5773	0,5759	0,5739	0,5760	5,9883	-50,00	0,5930	0,6075	0,6466	0,6693	0,6990
0,7150	0,7131	0,7116	0,6982	-4,0755	34,00	0,7261	0,7369	0,7650	0,7809	0,8013
0,8186	0,8166	0,8156	0,8204	4,2653	-15,00	0,8253	0,8324	0,8508	0,8609	0,8738
0,8912	0,8893	0,8890	0,8815	-1,1824	6,00	0,8942	0,8985	0,9094	0,9152	0,9225
0,9386	0,9369	0,9372	0,9426	1,95646	0	0,9392	0,9415	0,9471	0,9501	0,9539
0,9675	0,9661	0,9666	0,9648	0,5687	1,00	0,9668	0,9678	0,9703	0,9716	0,9733
0,9838	0,9829	0,9834	0,9871	1,1049	1,00	0,9827	0,9831	0,9839	0,9844	0,9849
0,9925	0,9918	0,9923	0,9926	0,9717	1,00	0,9914	0,9915	0,9916	0,9916	0,9917
0,9967	0,9963	0,9966	0,9982	1,0000	1,00	0,9959	0,9959	0,9957	0,9956	0,9955
0,9987	0,9984	0,9986	0,9990	1,0000	1,00	0,9982	0,9981	0,9979	0,9978	0,9976
0,9995	0,9994	0,9995	0,9999	1,0000	1,00	0,9992	0,9991	0,9990	0,9989	0,9987
0,9998	0,9998	0,9998	0,9999	1,0000	1,00	0,9997	0,9996	0,9995	0,9994	0,9994
0,9999	0,9999	0,9999	1,0000	1,0000	1,00	0,9999	0,9999	0,9998	0,9998	0,9997
1,0000	1,0000	1,0000	1,0000	1,0000	1,00	0,9999	1,0000	0,9999	1,0000	0,9998
1,0000	1,0000	1,0000	1,0000	1,0000	1,00	1,0000	1,0000	1,0000	1,0000	1,0000

4. táblázat. Az explicit és implicit Euler-módszer alkalmazása a $T=7$ időpillanatban különböző lépésközökkel ($\times 10^{-6}$ mol/cm³).

4.2.5. Crank-Nicolson-módszer

Vegyük az explicit és implicit Euler-módszer számtani közepét

$$c^{(k+1)} = c^{(k)} + 0,5 \cdot L_{mod} c^{(k)} + 0,5 \cdot L_{mod} \cdot c^{(k+1)},$$

egy oldalra rendezve a megfelelő iterációs tagokat

$$(I - 0,5 \cdot L_{mod}) \cdot c^{(k+1)} = c^{(k)} + 0,5 \cdot L_{mod} \cdot c^{(k)},$$

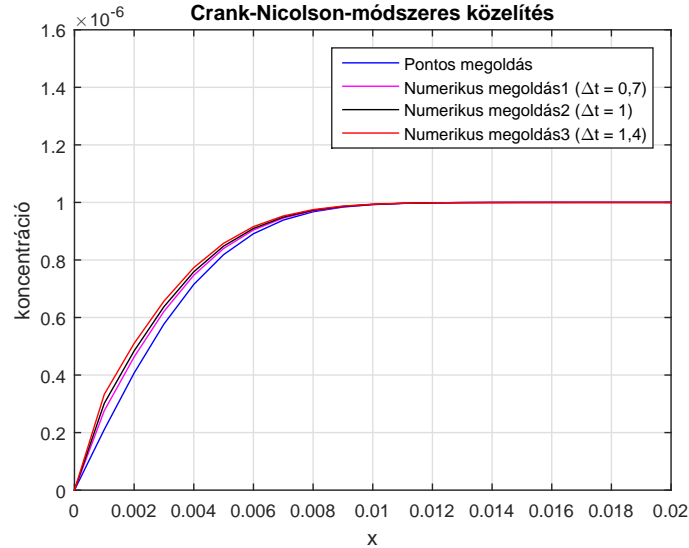
amelyet kifejezve $c^{(k+1)}$ -re kapjuk az alábbi összefüggést

$$c^{(k+1)} = (I - 0,5 \cdot L_{mod})^{-1} (I + 0,5 \cdot L_{mod}) \cdot c^{(k)}. \quad (4.2.13)$$

A Crank-Nicolson-módszerhez tartozó lépésmátrix:

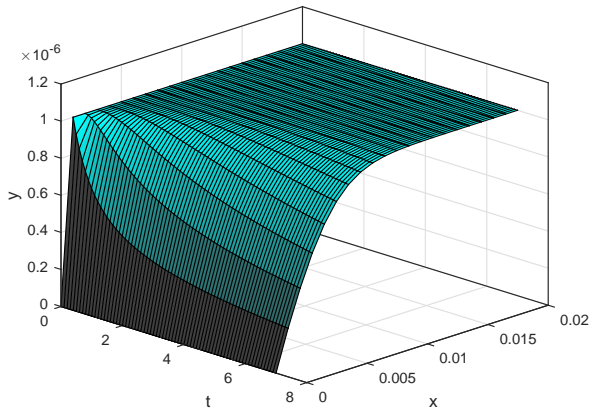
$$S := (I - 0,5 \cdot L_{mod})^{-1}(I + 0,5 \cdot L_{mod})$$

A 21. ábrán a Crank-Nicolson-módszer alkalmazása látható. Ez az eljárás is A -stabil, tehát a bemenő paraméterekkel arányosan változik a közelítés és a közelítés nem kezd el oszcillálni. Ez az eljárás jobb megoldást ad, mint az implicit Euler-módszer.

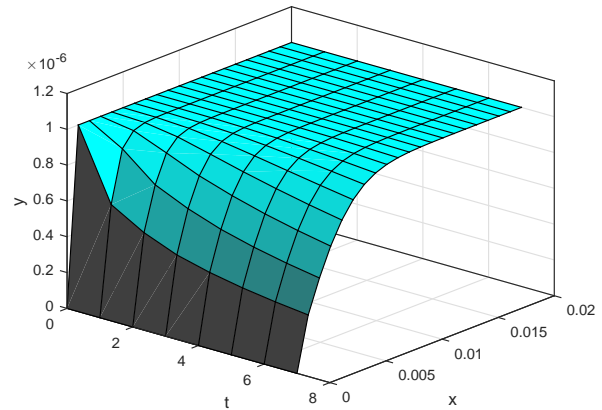


21. ábra. A Crank Nicolson-módszerrel történő közelítés $\Delta t = 0,7$, 1 , és $1,4$ lépésközökkel a $T=7$ időpillanatban.

A háromdimenziós ábrán is látszik, hogy nagy lépésközök esetén is jól viselkedik a módszer.



22. ábra. A Crank-Nicolson-módszerrel történő közelítés $\Delta t = 0,1$ esetén.



23. ábra. A Crank-Nicolson-módszerrel történő közelítés $\Delta t = 1$ esetén.

Összevetve az 5. táblázatot a 4. táblázattal elmondható, hogy kis lépésközök esetén az explicit Euler-módszer jobb értékeket ad a Crank-Nicolson módszernél, azonban nagyobb lépésközök esetén a Crank-Nicolson-módszer teljesít a legjobban. Stabilitás szempontjából is jobb ez a módszer az első kettőnél, hiszen növelve a lépésközt, kevesebbet torzulnak az értékek, mint az explicit és az implicit Euler-módszer esetében.

Pontos	Crank-Nicolson-módszer				
	h=0,1	h=0,2	h=0,5	h=0,7	h=1
0	0	0	0	0	0
0,2107	0,2211	0,2307	0,2585	0,2763	0,3017
0,4070	0,4162	0,4244	0,4480	0,4628	0,4838
0,5773	0,5845	0,5911	0,6098	0,6214	0,6376
0,7150	0,7198	0,7247	0,7385	0,7470	0,7589
0,8186	0,8210	0,8244	0,8341	0,8399	0,8482
0,8912	0,8918	0,8941	0,9003	0,9042	0,9095
0,9386	0,9381	0,9395	0,9433	0,9460	0,9490
0,9675	0,9665	0,9673	0,9695	0,9708	0,9727
0,9838	0,9828	0,9832	0,9844	0,9851	0,9861
0,9925	0,9916	0,9918	0,9924	0,9928	0,9933
0,9967	0,9961	0,9962	0,9965	0,9967	0,9969
0,9987	0,9983	0,9983	0,9985	0,9985	0,9986
0,9995	0,9993	0,9993	0,9994	0,9994	0,9994
0,9998	0,9997	0,9997	0,9997	0,9997	0,9997
0,9999	0,9999	0,9999	0,9999	0,9999	0,9999
1,0000	1,0000	1,0000	1,0000	1,0000	1,0000
1,0000	1,0000	1,0000	1,0000	1,0000	1,0000

5. táblázat. A Crank-Nicolson-módszer alkalmazása a $T=7$ időpillanatban különböző lépésközökkel ($\times 10^{-6}$ mol/cm³).

4.2.6. Runge-Kutta típusú módszerek

Eddig a

$$c^{(k+1)} = \exp(L_{mod}) \cdot c^{(k)},$$

egyenletben szereplő exponenciális kifejezést az explicit, implicit Euler-módszerrel és a Crank-Nicolson-módszerrel közelítettük. Most egy másik módszer családot alkalmazunk, a Runge-Kutta típusú módszereket. A 4.2.9 iterációban szereplő exponenciális kifejezés közelítéséhez sorbafejtést alkalmazunk.

$$S := \exp(L_{mod}) \approx I + L_{mod} + \frac{1}{2} \cdot L_{mod}^2 \quad (4.2.14)$$

$$S := \exp(L_{mod}) \approx I + L_{mod} + \frac{1}{2} \cdot L_{mod}^2 + \frac{1}{6} \cdot L_{mod}^3 \quad (4.2.15)$$

$$S := \exp(L_{mod}) \approx I + L_{mod} + \frac{1}{2} \cdot L_{mod}^2 + \frac{1}{6} \cdot L_{mod}^3 + \frac{1}{24} \cdot L_{mod}^4 \quad (4.2.16)$$

4.2.14-4.2.16-ból a következő iterációkat kapjuk

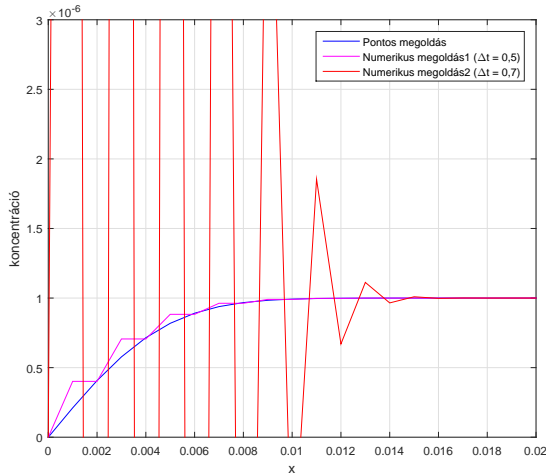
$$c^{(k+1)} = \left(I + L_{mod} + \frac{1}{2} \cdot L_{mod}^2 \right) \cdot c^{(k)},$$

$$c^{(k+1)} = \left(I + L_{mod} + \frac{1}{2} \cdot L_{mod}^2 + \frac{1}{6} \cdot L_{mod}^3 \right) \cdot c^{(k)},$$

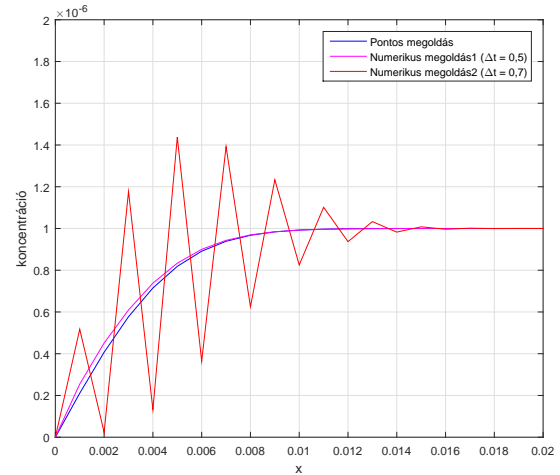
$$c^{(k+1)} = \left(I + L_{mod} + \frac{1}{2} \cdot L_{mod}^2 + \frac{1}{6} \cdot L_{mod}^3 + \frac{1}{24} \cdot L_{mod}^4 \right) \cdot c^{(k)},$$

amelyek rendre a másodrendű, harmadrendű és negyedrendű explicit Runge-Kutta típusú módszereket adják.

A 24. ábrán a másodrendű explicit Runge-Kutta módszer látható. Az explicit Runge-Kutta módszerek nem A -stabilak, ezért hasonló viselkedés figyelhető meg, mint az explicit Euler-módszer esetében. Ez az eljárás sokkal rosszabb értékeket ad, mint az explicit Euler-módszer.

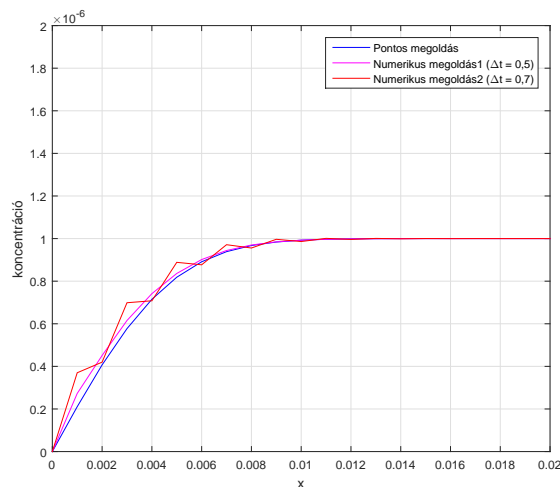


24. ábra. A másodrendű explicit Runge-Kutta módszer alkalmazása $\Delta t = 0,5$ és $0,7$ lépésközökkel a $T=7$ időpillanatban.



25. ábra. A harmadrendű explicit Runge-Kutta módszer alkalmazása $\Delta t = 0,5$ és $0,7$ lépésközökkel a $T=7$ időpillanatban.

A harmadrendű Runge-Kutta módszer már csak $\Delta t > 0,5$ esetén oszcillál, annál kisebb lépésköz esetén jól közelíti a megoldást, míg a negyedrendű Runge-Kutta még jobb közelítést ad. Azonban ezeknek a módszereknek nem csak az a hátrányuk, hogy nem A -stabilak, hanem minden egyes továbbfejtéssel nagyon megnő a műveletigény.



26. ábra. A negyedrendű explicit Runge-Kutta módszer alkalmazása $\Delta t = 0,5$ és $0,7$ lépésközökkel a $T=7$ időpillanatban.

A 6. és 7. táblázatok az explicit Runge-Kutta módszerek értékeit tartalmazzák különböző lépésközök esetén. A Crank-Nicolson-módszernél nem adnak jobb eredményt, csak kisebb lépésközök esetén jobbak, mint az implicit Euler módszer. $\Delta t > 0,5$ esetén hasonlóan viselkednek, mint az explicit Euler-módszer. Ezután hasonlóan olyan módszereket fogunk tekinteni, amelyeknek nagyobb a műveletigénye, azonban jobb közelítéseket kaphatunk. Ezek a lineáris többlépéses módszerek lesznek, azon belül is az Adams-Bashforth és Adams-Moulton módszerek.

Pontos	Másodrendű Runge-Kutta					Harmadrendű Runge-Kutta				
	h=0,1	h=0,2	h=0,5	h=0,7	h=1	h=0,1	h=0,2	h=0,5	h=0,7	h=1
0	0	0	0	0	0	0	0	0	0	0
0,2107	0,2224	0,2368	0,4013	27,8	1348,6	0,2214	0,2316	0,2547	0,5173	-1525,7
0,4070	0,4168	0,4267	0,4013	-35,9	-1854,2	0,4164	0,4251	0,4512	0,0237	2371,8
0,5773	0,5850	0,5934	0,7061	42,8	1974,2	0,5847	0,5915	0,6095	1,1771	-2751,3
0,7150	0,7201	0,7263	0,7061	-37,8	-1646,5	0,7199	0,7250	0,7381	0,1323	2696,1
0,8186	0,8213	0,8255	0,8831	32,1	1157,1	0,8211	0,8246	0,8338	1,4375	-2293,7
0,8912	0,8920	0,8947	0,8831	-21,3	-685,7	0,8919	0,8942	0,9001	0,3641	1734,6
0,9386	0,9382	0,9399	0,9625	15,2	349,3	0,9381	0,9395	0,9431	1,3919	-1171,8
0,9675	0,9665	0,9675	0,9625	-7,2	-149,3	0,9665	0,9673	0,9693	0,6249	713,9
0,9838	0,9828	0,9833	0,9903	5,2	55,7	0,9828	0,9832	0,9843	1,2333	-388,9
0,9925	0,9916	0,9919	0,9903	-1,0	-15,6	0,9916	0,9919	0,9924	0,8255	192,6
0,9967	0,9961	0,9963	0,9980	1,8	5,1	0,9961	0,9962	0,9965	1,1015	-83,4
0,9987	0,9983	0,9984	0,9980	0,7	0,2	0,9983	0,9983	0,9985	0,9370	34,2
0,9995	0,9993	0,9993	0,9997	1,1	1,1	0,9993	0,9993	0,9994	1,0329	-10,6
0,9998	0,9997	0,9997	0,9997	0,7	0,9	0,9997	0,9997	0,9997	0,9827	4,6
0,9999	0,9999	0,9999	1,0000	1,0	1,0	0,9999	0,9999	0,9999	1,0080	0,05
1,0000	1,0000	1,0000	1,0000	0,9	1,0	1,0000	1,0000	1,0000	0,9964	1,2
1,0000	1,0000	1,0000	1,0000	1,0	1,0	1,0000	1,0000	1,0000	1,0014	0,9

6. táblázat. A másod- és harmadrendű Runge-Kutta típusú módszer alkalmazása a $T=7$ időpillanatban különböző lépésközökkel ($\times 10^{-6}$ mol/cm³).

Pontos	Negyedrendű Runge-Kutta				
	h=0,1	h=0,2	h=0,5	h=0,7	h=1
0	0	0	0	0	0
0,2107	0,2214	0,2322	0,2729	0,3700	674,31
0,4070	0,4164	0,4251	0,4511	0,4198	-977,67
0,5773	0,5847	0,5917	0,6154	0,6988	1177,99
0,7150	0,7199	0,7251	0,7414	0,7079	-1193,08
0,8186	0,8211	0,8247	0,8363	0,8885	1081,41
0,8912	0,8919	0,8942	0,9016	0,8769	-882,31
0,9386	0,9381	0,9396	0,9440	0,9713	661,54
0,9675	0,9665	0,9673	0,9698	0,9557	-453,23
0,9838	0,9828	0,9833	0,9846	0,9966	289,21
0,9925	0,9916	0,9919	0,9925	0,9861	-168,07
0,9967	0,9961	0,9962	0,9966	1,0011	92,76
0,9987	0,9983	0,9983	0,9985	0,9961	-45,07
0,9995	0,9993	0,9993	0,9994	1,0008	22,38
0,9998	0,9997	0,9997	0,9997	0,9990	-8,15
0,9999	0,9999	0,9999	0,9999	1,0003	4,61
1,0000	1,0000	1,0000	1,0000	0,9997	0,31
1,0000	1,0000	1,0000	1,0000	1,0001	1,43

7. táblázat. A negyedrendű Runge-Kutta típusú módszer alkalmazása a $T=7$ időpillanatban különböző lépésközökkel ($\times 10^{-6}$ mol/cm³).

4.2.7. Lineáris többlépéses módszerek

Most olyan módszereket fogunk alkalmazni, amelyekben a következő időbeli vektor kiszámításához több közvetlen megelőző vektort használunk, nemcsak egyet. Ennél a módszer családnál is vizsgálni fogom külön az explicit és implicit típusokat. Tekintsük az általános többlépéses módszert leíró képletet

$$\frac{a_0 y_n + a_1 y_{n-1} + \dots + a_m y_{n-m}}{h} = b_0 f_n + b_1 f_{n-1} + \dots + b_m f_{n-m},$$

ezen belül is az Adams-típusú módszereket ($a_0 = 1, a_1 = -1, a_2 = 0, \dots, a_n = 0$), továbbá az explicit esetet, vagyis $b_0 = 0$, azaz az Adams-Bashforth módszereket.

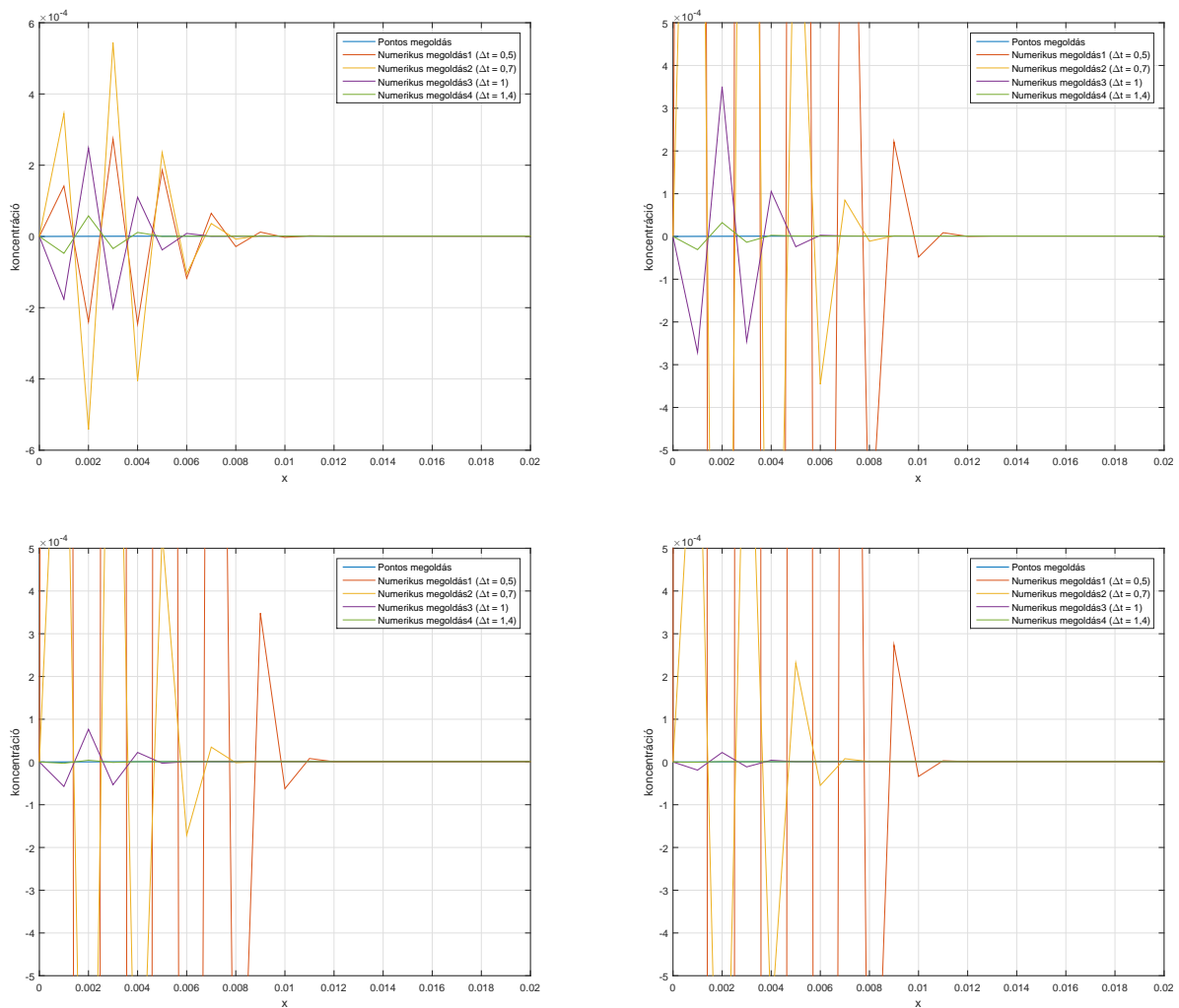
Adams-Bashforth módszerek

A következő együtthatókkal rendelkező eljárásokat fogom vizsgálni, amelyet a 8. táblázat foglal össze.

Jel	b_0	b_1	b_2	b_3	b_4	b_5	b_6
AB_1	0	1					
AB_2	0	$3/2$	$-1/2$				
AB_3	0	$23/12$	$-16/12$	$5/12$			
AB_4	0	$55/24$	$-59/24$	$37/24$	$-9/24$		
AB_5	0	$1901/720$	$-2774/720$	$2616/720$	$-1274/720$	$251/720$	
AB_6	0	$4277/720$	$-7923/720$	$9982/720$	$-7298/720$	$2877/720$	$-475/720$

8. táblázat. Adams-Bashforth módszerek együtthatói.

$$y_n - y_{n-1} = \Delta t \left[\frac{3}{2}f_{n-1} - \frac{1}{2}f_{n-2} \right] \quad (4.2.17)$$



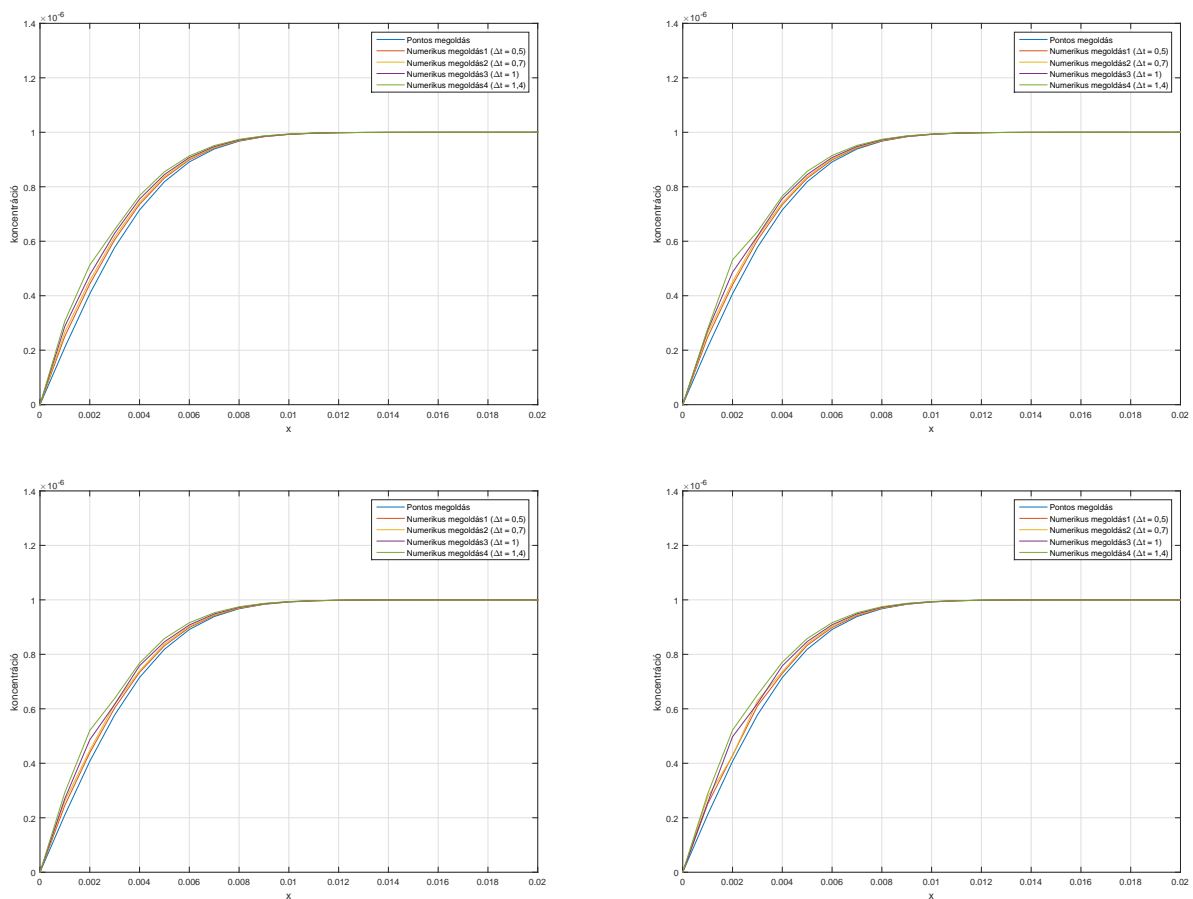
27. ábra. A kettő-, három-, négy-, és ötlépéses Adams-Bashforth módszerek.

Adams-Moulton módszerek

Jel	b_0	b_1	b_2	b_3	b_4	b_5
AM_{1a}	1	0				
AM_{1b}	1/2	1/2				
AM_2	5/12	8/12	-1/12			
AM_3	9/24	19/24	-5/24	1/24		
AM_4	251/720	646/720	-264/720	106/720	-19/720	
AM_5	475/1440	1427/1440	-798/1440	482/1440	-173/1440	27/1440

9. táblázat. Adams-Moulton módszerek együtthatói.

A fenti 9. táblázat tartalmazza az Adams-Moulton módszerek együtthatóit az ötlépéses eljárásig. Az alsó index a metódus lépésszámára utal. Vegyük észre, hogy az első két sor rendre az implicit Euler módszert és a Crank-Nicolson módszert írja le, ezt a két módszert már vizsgáltuk.



28. ábra. A kettő-, három-, négy-, és ötlépéses Adams-Moulton módszerek.

4.3. A módszerek összehasonlítása

A 10. táblázatban az utolsó időpillanatbeli hibavektorok normái szerepelnek különböző időbeli lépésköz esetén. Az egyetlen változás a 3. táblázatban szereplő paraméterekhez képest, hogy itt T , vagyis az időbeli végpont 7 mp helyett 8 mp.

Norma	EEM	IEM	CNM
No1 ($\Delta t = 0,4$)	$3,3271 \times 10^{-8}$	$3,0855 \times 10^{-7}$	$1,4627 \times 10^{-7}$
NoInf ($\Delta t = 0,4$)	$6,3255 \times 10^{-9}$	$7,5342 \times 10^{-8}$	$3,6620 \times 10^{-8}$
No2 ($\Delta t = 0,4$)	$1,2289 \times 10^{-8}$	$1,2886 \times 10^{-7}$	$6,1296 \times 10^{-8}$
No1 ($\Delta t = 0,2$)	$1,8161 \times 10^{-8}$	$1,6071 \times 10^{-7}$	$7,4416 \times 10^{-8}$
NoInf ($\Delta t = 0,2$)	$3,0174 \times 10^{-9}$	$3,8730 \times 10^{-8}$	$1,8808 \times 10^{-8}$
No2 ($\Delta t = 0,2$)	$6,3877 \times 10^{-9}$	$6,7078 \times 10^{-8}$	$3,1481 \times 10^{-8}$
No1 ($\Delta t = 0,1$)	$1,3266 \times 10^{-8}$	$8,2684 \times 10^{-8}$	$3,9426 \times 10^{-8}$
NoInf ($\Delta t = 0,1$)	$1,7287 \times 10^{-9}$	$1,9788 \times 10^{-8}$	$9,7067 \times 10^{-9}$
No2 ($\Delta t = 0,1$)	$4,1097 \times 10^{-9}$	$3,4389 \times 10^{-8}$	$1,6247 \times 10^{-8}$

10. táblázat. Az utolsó időpillanati hibavektorok normái.

Ahogy az kivehető 10 táblázatból, az explicit és implicit Euler-módszerek megfelelő normái az időbeli lépésköz finomításával - jelen esetben felezésével - szintén feleződik. A másodrendű Crank-Nicolson-módszerrel szembeni elvárásunk, miszerint a hiba negyedelődik az időlépés felezésének hatására nem teljesül, ugyanis a térbeli diszkretizációs paraméter értékét nem változtattuk, csupán az időbeli közelítést finomítottuk.

Norma	RK2	RK3	RK4
No1 ($\Delta t = 0,4$)	$2,2924 \times 10^{-7}$	$1,4912 \times 10^{-7}$	$1,6344 \times 10^{-7}$
NoInf ($\Delta t = 0,4$)	$7,5037 \times 10^{-8}$	$3,6407 \times 10^{-8}$	$4,3639 \times 10^{-8}$
No2 ($\Delta t = 0,4$)	$1,0118 \times 10^{-7}$	$6,2647 \times 10^{-8}$	$6,9076 \times 10^{-8}$
No1 ($\Delta t = 0,2$)	$8,8375 \times 10^{-8}$	$7,6996 \times 10^{-8}$	$7,8184 \times 10^{-8}$
NoInf ($\Delta t = 0,2$)	$2,4463 \times 10^{-8}$	$1,9669 \times 10^{-8}$	$2,0259 \times 10^{-8}$
No2 ($\Delta t = 0,2$)	$3,7929 \times 10^{-8}$	$3,2657 \times 10^{-8}$	$3,3203 \times 10^{-8}$
No1 ($\Delta t = 0,1$)	$4,2424 \times 10^{-8}$	$4,0215 \times 10^{-8}$	$4,0328 \times 10^{-8}$
NoInf ($\Delta t = 0,1$)	$1,0909 \times 10^{-8}$	$1,0002 \times 10^{-8}$	$1,0059 \times 10^{-8}$
No2 ($\Delta t = 0,1$)	$1,7651 \times 10^{-8}$	$1,6614 \times 10^{-8}$	$1,6667 \times 10^{-8}$

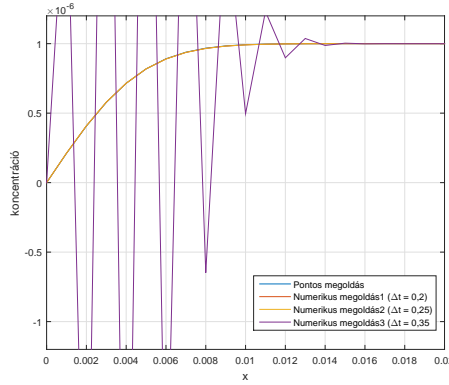
11. táblázat. Az utolsó időpillanati hibavektorok normái a Runge-Kutta módszerek esetén.

A Runge-Kutta típusú módszereknél is ugyanez mondható el, az időbeli lépésközök felezésével a hiba itt is feleződik.

Norma	kétlépéses	háromlépéses	négylépéses	ötlépéses
No1 ($\Delta t = 0,4$)	$8,9680 \times 10^{-4}$	1,4798	19,4672	$2,7053 \times 10^2$
NoInf ($\Delta t = 0,4$)	$1,5118 \times 10^{-4}$	0,2829	3,7179	58,0006
No2 ($\Delta t = 0,4$)	$3,1798 \times 10^{-4}$	0,5575	7,3519	$1,0783 \times 10^2$
No1 ($\Delta t = 0,2$)	$1,2120 \times 10^{-8}$	0,0011	79,7807	$1,9604 \times 10^6$
NoInf ($\Delta t = 0,2$)	$1,5138 \times 10^{-9}$	$1,5819 \times 10^{-4}$	13,5891	$3,1277 \times 10^5$
No2 ($\Delta t = 0,2$)	$3,5912 \times 10^{-9}$	$3,6897 \times 10^{-4}$	28,5105	$6,7207 \times 10^5$
No1 ($\Delta t = 0,1$)	$1,1794 \times 10^{-8}$	$1,1590 \times 10^{-8}$	$3,3731 \times 10^{-4}$	$7,1746 \times 10^6$
NoInf ($\Delta t = 0,1$)	$1,6330 \times 10^{-9}$	$1,5629 \times 10^{-9}$	$3,8909 \times 10^{-5}$	$8,7591 \times 10^5$
No2 ($\Delta t = 0,1$)	$3,5939 \times 10^{-9}$	$3,4792 \times 10^{-9}$	$9,8218 \times 10^{-5}$	$2,1465 \times 10^6$

12. táblázat. Az utolsó időpillanati hibavektorok normái az Adams-Bashforth módszerek esetén.

A 12. táblázat a magáért beszél, az Adams-Bashforth módszerek hibáinak a normái teljesen rendszertelenül viselkednek, össze-vissza ingadoznak az értékek. Bár azt érdemes megjegyezni és kiemelni, hogy $\Delta t = 0,25$ lépésközig a kétlépéses Adams-Bashforth módszer kisebb hibát ad, mint az explicit Euler-módszer, de annál nagyobb lépésköz esetén sokkal jobban romlanak az értékei.



29. ábra. A kétlépéses Adams-Bashforth módszer kritikus működése.

Ahogy a 29. ábra is szemlélteti, $\Delta t = 0,25$ esetén a közelítés „rásimul” a pontos megoldásra, azonban 0,1-del nagyobb lépésköz esetén már erősen oszcillálni kezd.

Norma	kétlépéses	háromlépéses	négylépéses	ötlépéses
No1 ($\Delta t = 0,4$)	$1,0000 \times 10^{-11}$	$1,0000 \times 10^{-11}$	$1,0000 \times 10^{-11}$	$1,0027 \times 10^{-11}$
NoInf ($\Delta t = 0,4$)	$1,0000 \times 10^{-11}$	$1,0000 \times 10^{-11}$	$9,9999 \times 10^{-12}$	$1,0027 \times 10^{-11}$
No2 ($\Delta t = 0,4$)	$1,0000 \times 10^{-11}$	$1,0000 \times 10^{-11}$	$9,9999 \times 10^{-12}$	$1,0027 \times 10^{-11}$
No1 ($\Delta t = 0,2$)	$4,9999 \times 10^{-12}$	$4,9999 \times 10^{-12}$	$5,0000 \times 10^{-12}$	$5,0138 \times 10^{-12}$
NoInf ($\Delta t = 0,2$)	$4,9999 \times 10^{-12}$	$4,9999 \times 10^{-12}$	$5,0000 \times 10^{-12}$	$5,0138 \times 10^{-12}$
No2 ($\Delta t = 0,2$)	$4,9999 \times 10^{-12}$	$4,9999 \times 10^{-12}$	$5,0000 \times 10^{-12}$	$5,0138 \times 10^{-12}$
No1 ($\Delta t = 0,1$)	$2,4999 \times 10^{-12}$	$2,4999 \times 10^{-12}$	$2,5000 \times 10^{-12}$	$2,5069 \times 10^{-12}$
NoInf ($\Delta t = 0,1$)	$2,4999 \times 10^{-12}$	$2,4999 \times 10^{-12}$	$2,4999 \times 10^{-12}$	$2,5069 \times 10^{-12}$
No2 ($\Delta t = 0,1$)	$2,4999 \times 10^{-12}$	$2,4999 \times 10^{-12}$	$2,4999 \times 10^{-12}$	$2,5069 \times 10^{-12}$

13. táblázat. Az utolsó időpillanati hibavektorok normái az Adams-Bashforth módszerek esetén, ha a pontos megoldással indítjuk a módszert.

Kiindulva abból, hogyha már előállítottuk az analitikus megoldást, akkor azt használjuk fel a kezdeti lépésekhez és nem a Crank-Nicolson-módszert. Elég biztató értékeket kapunk a normákra vonatkozóan, itt tisztán látszik a feleződés, a lépésközök felezésével, míg amikor a Crank-Nicolson-módszert használtuk fel, akkor az értékekből nem lehetett ezt leszűrni.

Norma	kétlépéses	háromlépéses	négylépéses	ötlépéses
No1 ($\Delta t = 0,4$)	$1,2424 \times 10^{-7}$	$1,1514 \times 10^{-7}$	$1,1070 \times 10^{-7}$	$1,0894 \times 10^{-7}$
NoInf ($\Delta t = 0,4$)	$3,0936 \times 10^{-8}$	$2,8193 \times 10^{-8}$	$2,6564 \times 10^{-8}$	$2,7150 \times 10^{-8}$
No2 ($\Delta t = 0,4$)	$5,2133 \times 10^{-8}$	$4,8064 \times 10^{-8}$	$4,5918 \times 10^{-8}$	$4,5233 \times 10^{-8}$
No1 ($\Delta t = 0,2$)	$6,2909 \times 10^{-8}$	$5,7737 \times 10^{-8}$	$5,4872 \times 10^{-8}$	$5,3081 \times 10^{-8}$
NoInf ($\Delta t = 0,2$)	$1,5838 \times 10^{-8}$	$1,4386 \times 10^{-8}$	$1,3499 \times 10^{-8}$	$1,2892 \times 10^{-8}$
No2 ($\Delta t = 0,2$)	$2,6586 \times 10^{-8}$	$2,4298 \times 10^{-8}$	$2,2976 \times 10^{-8}$	$2,2131 \times 10^{-8}$
No1 ($\Delta t = 0,1$)	$3,4164 \times 10^{-8}$	$3,1677 \times 10^{-8}$	$3,0172 \times 10^{-8}$	$2,9157 \times 10^{-8}$
NoInf ($\Delta t = 0,1$)	$8,1905 \times 10^{-9}$	$7,4421 \times 10^{-9}$	$6,9773 \times 10^{-9}$	$6,6542 \times 10^{-9}$
No2 ($\Delta t = 0,1$)	$1,3765 \times 10^{-8}$	$1,2574 \times 10^{-8}$	$1,1857 \times 10^{-8}$	$1,1375 \times 10^{-8}$

14. táblázat. Az utolsó időpillanati hibavektorok normái az Adams-Moulton módszerek esetén.

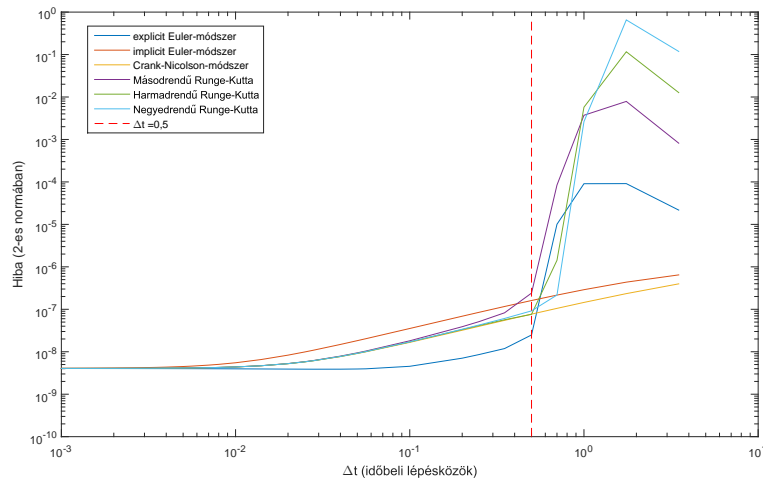
Az Adams-Moulton módszerek esetén már jobb a helyzet, ugyanaz a hibafeleződés mondható el, mint eddig. Ebben az esetben is az Crank-Nicolson-módszert használtuk fel kezdeti lépések kiszámítására. Ennél a módszernél is próbáljunk egy jobb közelítést adni úgy, hogy itt is a pontos megoldást alkalmazzuk kezdeti lépések megadására.

Norma	kétlépéses	háromlépéses	négylépéses	ötlépéses
No1 ($\Delta t = 0,4$)	$9,9999 \times 10^{-12}$	$9,9999 \times 10^{-12}$	$9,9999 \times 10^{-12}$	$3,9998 \times 10^{-11}$
NoInf ($\Delta t = 0,4$)	$9,9999 \times 10^{-12}$	$9,9999 \times 10^{-12}$	$9,9999 \times 10^{-12}$	$3,9998 \times 10^{-11}$
No2 ($\Delta t = 0,4$)	$9,9999 \times 10^{-12}$	$9,9999 \times 10^{-12}$	$9,9999 \times 10^{-12}$	$3,9998 \times 10^{-11}$
No1 ($\Delta t = 0,2$)	$4,9999 \times 10^{-12}$	$4,9999 \times 10^{-12}$	$4,9999 \times 10^{-12}$	$1,9999 \times 10^{-11}$
NoInf ($\Delta t = 0,2$)	$4,9999 \times 10^{-12}$	$4,9999 \times 10^{-12}$	$4,9999 \times 10^{-12}$	$1,9999 \times 10^{-11}$
No2 ($\Delta t = 0,2$)	$4,9999 \times 10^{-12}$	$4,9999 \times 10^{-12}$	$4,9999 \times 10^{-12}$	$1,9999 \times 10^{-11}$
No1 ($\Delta t = 0,1$)	$2,4999 \times 10^{-12}$	$2,4999 \times 10^{-12}$	$2,4999 \times 10^{-12}$	$9,9999 \times 10^{-12}$
NoInf ($\Delta t = 0,1$)	$2,4999 \times 10^{-12}$	$2,4999 \times 10^{-12}$	$2,4999 \times 10^{-12}$	$9,9999 \times 10^{-12}$
No2 ($\Delta t = 0,1$)	$2,4999 \times 10^{-12}$	$2,4999 \times 10^{-12}$	$2,4999 \times 10^{-12}$	$9,9999 \times 10^{-12}$

15. táblázat. Az utolsó időpillanati hibavektorok normái az Adams-Moulton módszerek esetén, ha a pontos megoldással indítjuk a módszert.

A 15. táblázatban szereplő normák nagyon hasonlóak a 13. táblázatban kapottakhoz, itt is pontosan ugyanaz mondható el.

Átfogóbb képet nyújtanak számunkra a következő ábrák, amelyek a lehetséges lépésközök esetén mutatják a hiba változását és remekül szemléltetik az egyes módszerek hatékonyságát.

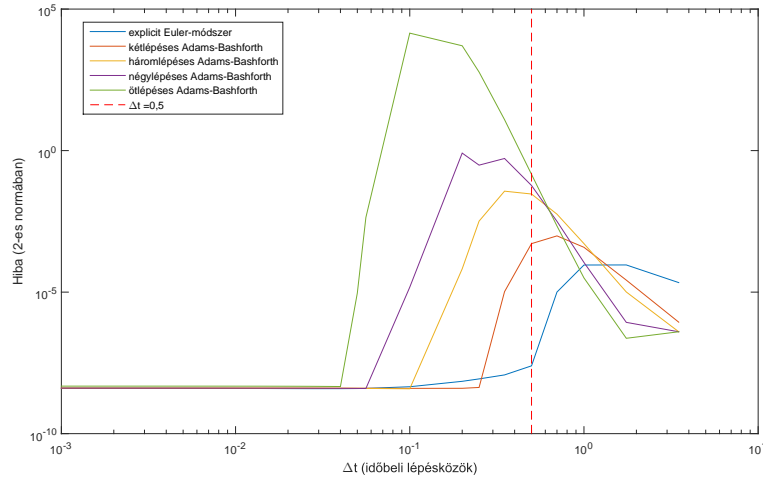


30. ábra. A numerikus módszerek hibája.

A 30. ábra az eddig vizsgált numerikus módszerek hibáit szemlélteti különféle lépésközök esetén. Az ábra pontosan azokat a tulajdonságokat reprezentálja, amik az eddigi vizsgálódások során mutatkoztak meg. Az explicit Euler-módszer teljesít a legjobban $\Delta t < 0,5$ lépésköz esetén, azután a hiba nagyon megugrik. Az implicit Euler-módszer és a Crank-Nicolson-módszer az A -stabilitásnak köszönhetően folytonosan változik a bemenő paraméterek megváltoztatása során. Az értékek nem szállnak el, ezek a módszerek viselkednek a legjobban nagy lépésköz esetén.

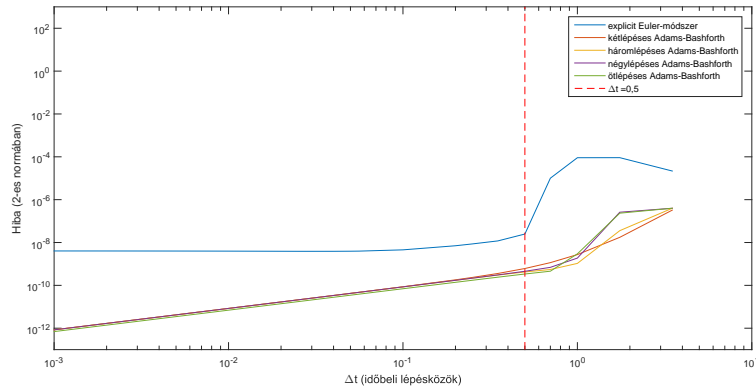
A Runge-Kutta típusú módszerek kis lépések estén az implicit Euler-módszer és a Crank-Nicolson-módszer között mozognak. Az egyetlen meglepő tulajdonság az eddigi eredményekhez viszonyítva, hogy extrém nagy lépésközök esetén (itt $\Delta t \geq 1$ esetekre kell gondolni) rendre a másodrendű, harmadrendű, negyedrendű Runge-Kutta módszerek adják a legnagyobb hibákat. Illetve még az is külön érdekesség, hogy az utolsó lehetséges (legnagyobb) lépésközöknél valamennyi javulás figyelhető meg a Runge-Kutta módszerek és az explicit Euler-módszer használata során.

Most az Adams-Bashforth és az Adams-Moulton módszereket fogjuk megvizsgálni, mindegyik módszernél azt a két ágat fogjuk összehasonlítani, hogy mit használtunk fel a kezdeti lépésekhez. Abban az esetben, amikor a Crank-Nicolson-módszert használtuk fel a kezdeti értéke meghatározásához nem számíthatunk olyan jó eredményekre, hiszen már alaptól plusz hibával terheltük a közelítésünket. Viszont, ha már előállítottuk a pontos megoldást és ezek szolgálnak kezdeti lépésként, akkor már nem terheltük alaptól hibával a közelítésünket, tehát jobb eredményeket kapunk.



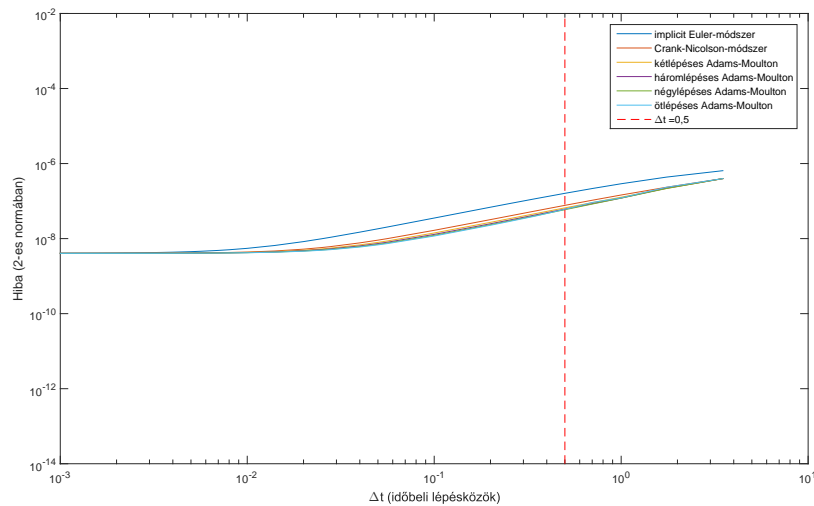
31. ábra. Az Adams-Bashforth módszerek hibája.

A 31. ábra is csak azt bizonyítja, amit eddig láttunk, az Adams-Bashforth módszerekkel zsákutcába futunk, ha a Crank-Nicolson-módszerrel határozzuk meg a kezdeti értékeket. Ebben az esetben nem érdemes ezt az eljárást alkalmaznunk, hiszen csak rosszabb közelítéseket állítottunk elő velük. A görbék viselkedése jól szemlélteti az explicit lineáris többlépéses módszerek egyik nagy hátrányát, hogy nem A -stabilak. Az ötlépéses módszer száll el a leggyorsabban, már $0,025$ -es lépésköz esetén megugrik a hiba és $\Delta t = 0,05$ esetén már eléri a $12,4113$ értéket (kezdetben $4,7739 \times 10^{-4}$), amivel a legrosszabb módszer közé került. A kétlépéses módszer a legjobb ezek közül, de $\Delta t = 0,35$ lépésköz esetén már az sem közelíti a megoldást.



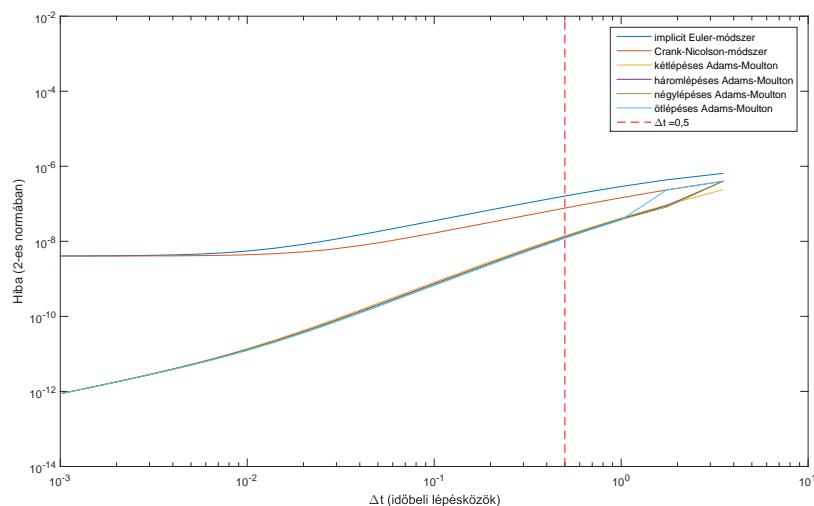
32. ábra. Az Adams-Bashforth módszerek hibája, ha a pontos megoldással indítjuk a módszert.

Mivel vizsgálataink során a térbeli diszkretizáció lépésközének értéke állandó volt, csak az időbeli diszkretizációs paraméter értékét változtattuk, a hibatáblázatokról láthattuk, hogy minden módszer elsőrendűként viselkedik. Ebből kifolyólag – és ahogy azt az explicit Euler-módszer esetéből is láthattuk – nem szabad elhanyagolni az explicit lineáris többlépéses módszereket, mert jobb közelítéseket kaptunk. Ahogy az a 32. ábrán is látható az eddigi eljárások közül ezek adják a legjobb közelítéseket, kis lépésköz esetén az eddigi 4×10^{-9} hiba lecsökkent 6×10^{-13} -ra. Noha a hiba most is viszonylag gyorsan nő és még nem is A -stabil módszer, de az elmondható, hogy ezek ellenére is pontosabb eredményeket kapunk, csak $\Delta t > 1$ lépésköz esetén adhat rosszabb eredményeket, ami elég nagy szám.



33. ábra. Az Adams-Moulton módszerek hibája.

Az Adams-Moulton módszerek jobb közelítéseket adnak, mint az egylépéses implicit módszerek. A kétlépésestől a ötlépéses Adams-Moulton módszerekig folyamatos javulás érzékelhető.



34. ábra. Az Adams-Moulton módszerek hibája, ha a pontos megoldással indítjuk a módszert.

Próbáljunk jobb közelítéseket elérni az Adams-Bashforth módszereknél használt elvvel. A 34. ábrán látható, hogy nagyjából hasonló sikert értünk el, mint az explicit lineáris többlépéses módszerek esetében. Azonban az explicit lineáris többlépéses módszerek kis lépésközök esetén jobbnak bizonyulnak 0,3-0,5-tel kisebb hibákat kapunk.

5. Összefoglalás

Szakdolgozatom célja egy elektorkémiai kísérlet modellezése, leírása, valamint vizsgálata volt. A Cottrell-kísérlet során a koncentráció változása Fick II. törvénye alapján leírható a következő diffúziós egyenlettel:

$$\frac{\partial c(x, t)}{\partial t} = D \cdot \frac{\partial^2 c(x, t)}{\partial x^2}.$$

Dolgozatomban levezettem ennek a parciális differenciálegyenletnek az analitikus megoldását, valamint 14+8 különböző numerikus közelítést alkalmaztam a koncentrációvektor értékeinek meghatározására.

A numerikus közelítések közül a lineáris többlépéses módszerek bizonyultak a legjobbnak – az Adams-Bashforth és Adams-Moulton – mégpedig abban az esetben, ha kezdeti értékeknek a pontos megoldás értékeit állítjuk be. Ellenkező esetben, ha kezdeti lépéseket is közelítéssel állítjuk elő, akkor plusz hibával terheljük a közelítésünket, abban az esetben ezek a módszerek nem teljesítenek jól. Ugyan jó közelítéseket kaptunk az egyik ágon, de fontos kihangsúlyozni, hogy ennek ára volt, hiszen ezeknek az eljárásoknak elég nagy a műveletigénye. A másik nagy hátránya ezeknek a metódusoknak, hogy nem A -stabilak, így nagyobb lépésköz esetén nagyobb hibákat vétenek.

A másik nagy módszercsalád az egylépéses eljárások voltak. Ezek közül az explicit Euler-módszer teljesít a legjobban kis lépésközök esetén, azonban $\Delta t > 0,5$ megválasztásokkal már nagyon rosszul viselkedik a közelítés amiatt, hogy a módszer nem A -stabil. Ezt követte a Crank-Nicolson-módszer, amelynek egyik fontos tulajdonsága az A -stabilitás volt. A Crank-Nicolson-módszer után a Runge-Kutta típusú módszerek teljesítettek a legjobban (csak kis lépésközök alkalmazásával) $\Delta t > 0,5$ esetén rosszabbul viselkedtek, mint az explicit Euler-módszer. Ezeknek az eljárásoknak több hátránya is volt, mivel az explicit Runge-Kutta típusú módszerek nem A -stabilak, így az előbbi lépésközökre vonatkozó egyenlőtlenség fennállása esetén nagyon rossz közelítéseket adtak. Végezetül az egylépéses módszerek közül ezeket követte az implicit Euler-módszer, amely A -stabil eljárás.

A legrosszabb közelítéseket az Adams-Bashforth módszerek adták abban az esetben, ha a kezdeti értékeket is közelítéssel állítottuk elő.

További vizsgálatok tárgyát képezheti az egyes numerikus módszerek összehasonlítása abban az esetben, ha a tér- és időbeli diszkretizációs paraméterek értékét együttesen változtatjuk, valamint vizsgálhatjuk a térbeli diszkretizáció pontosságának hatását a numerikus eredmények értékére. Ezen vizsgálatok túlmutatnak az alapszakos tanulmányokon, de későbbi munkák alapjául szolgálhatnak.

6. A szakdolgozatban szereplő Matlab kódok

A szakdolgozatban szereplő ábrákat a MATLAB és a www.draw.io programokkal készítettem. A legtöbb függvénynek annyifajta variánsa van, ahány numerikus módszert vizsgáltam a szakdolgozatomban. A szükségtelen hosszúság elkerülése érdekében nem töltöttem fel a függvények minden változatát teljesen, hanem egy verzióban a megváltozott kódblokkok után komment formájában beilleszttem az egyes függvényvariációkhoz tartozó kódrészleteket.

A Runge-Kutta típusú módszerek stabilitási tartományai - 8. ábra (4.7. A Runge-Kutta módszer)

```

1 %rkStabilityRegion
2 %Tengelyek beosztása es stabfv
3 xvector = linspace(-5,2,201);
4 yvector = linspace(-4,4,201);
5 [x,y] = meshgrid(xvector,yvector);
6 z = x + i*y;
7 %rk1stabfun = 1 + z;
8 %rk2stabfun = 1 + z + 0.5*z.^2;
9 %rk3stabfun = 1 + z + 0.5*z.^2 + (1/6)*z.^3;
10 rk4stabfun = 1 + z + 0.5*z.^2 + (1/6)*z.^3 + (1/24)*z.^4;
11 %rk1sr = abs(rk1stabfun);
12 %rk2sr = abs(rk2stabfun);
13 %rk3sr = abs(rk3stabfun);
14 rk4sr = abs(rk4stabfun);
15 %Kirajzoltatas
16 contourf(x,y,rk4sr,[0.001 1],'k');
17 colormap([1 0 0; 1 1 1]) %piros-feher
18 axis([-5,2,-4,4]);
19 axis('square');
20 xlabel('Re');
21 ylabel('Im');
22 grid on;

```

Hibafüggvények - 9-10. ábra (5.2. Az analitikus megoldás)

```

1 %A Gauss-fele hibafuggveny
2 close all
3
4 x1 = -2:0.1:2;
5 y2 = erf(x1);
6 plot(x1,y2)
7 grid on
8 title('Az Gauss-fele hibafuggveny a [-2,2] intervallumon')
9 xlabel('x')
10 ylabel('erf(x)')

```

```

1 %A komplementer hibafuggveny
2 close all
3
4 x2 = -2:0.1:2;
5 z = erf(x2);
6 y2 = 1-z;
7 plot(x2,y2)
8 grid on
9 title('A komplementer hibafuggveny a [-2,2] intervallumon')
10 xlabel('x')
11 ylabel('erfc(x)')

```

A pontos megoldás grafikonjai - 11-12. ábra (5.2. Az analitikus megoldás)

```

1 %3D-s abra
2 function [Accurate, Accurate2] = accSolution(n, deltat, deltax, con, T)
3 D=10^(-6);
4
5 Accurate = zeros(n+1, (T/deltat)+1);
6 for i = 1:(n+1)
7     for j = 1:(T/deltat)+1
8         if j== 1
9             Accurate(i, :) = 1;
10        else
11            auxVar = deltax*(i-1)/(sqrt(4*D*(j-1)*deltat));
12            Accurate(i, j) = erf(auxVar);
13        end
14    end
15 end
16 Accurate2 = con*Accurate
17
18 Tmtx = zeros(n+1, (T/deltat)+1);
19 for i = 1:n+1
20     for j = 1:(T/deltat)+1
21         Tmtx(:, j) = deltat*(j-1);
22     end
23 end
24 Tmtx
25
26 Xmtx = zeros(n+1, (T/deltat)+1);
27 for i = 1:n+1
28     for j = 1:(T/deltat)+1
29         Xmtx(i, :) = deltax*(i-1);
30     end
31 end
32 Xmtx
33
34 mesh(Tmtx, Xmtx, Accurate2)
35 xlabel('t');
36 ylabel('x');
37 zlabel('y');
38 grid on;
39
40 end

```

```

1 %2D-s abra
2 function [xmtx, value] = accSolution2D(n, deltat, deltax, con, T)
3 D=10^(-6);
4
5 tmtx = zeros(1, (T/deltat)+1);
6 for i = 1:(T/deltat)+1
7     tmtx(i) = deltat*(i-1);
8 end
9 tmtx
10
11 xmtx = zeros(1, n+1);
12 for i = 1:(n+1)
13     xmtx(i) = deltax*(i-1);
14 end
15
16 Accurate = zeros(n+1, (T/deltat)+1);
17 Accurate2 = zeros(n+1, (T/deltat)+1);
18 value = zeros(1, n+1);
19 for i = 1:(n+1)
20     for j = 1:(T/deltat)+1
21         if j== 1
22             Accurate(i, :) = 0;
23         else

```

```

24     auxVar = deltax*(i-1)/(sqrt(4*D*(j-1)*deltat));
25     Accurate(i,j) = erf(auxVar);
26     end
27 end
28 end
29 Accurate2 = con*Accurate;
30 Accurate2
31 value = Accurate2(:,(T/deltat)+1)';
32
33 plot(xmtx,value)
34 grid on
35 xlabel('x');
36 ylabel('koncentracio');
37
38
39 end

```

EEM, IEM, CNM és RK közelítések - 14-26. ábra (5.3.3. Az *explicit Euler-módszer*) - (5.3.6. A *Runge-Kutta típusú módszerek*)

```

1 function [yMtx,Tmtx,Xmtx] = szimulacio(n,D,deltat,deltax,con,T)
2
3     Lmtx = zeros(n+1,n+1);
4     for i = 1:n+1
5         for j = 1:n+1
6             if ((i == 1 && j == 1) || (i == n+1 && j == n+1))
7                 Lmtx(i,j) = -1;
8             elseif i==j
9                 Lmtx(i,j) = -2;
10            elseif i == j-1 || i == j+1
11                Lmtx(i,j) = 1;
12            end
13        end
14    end
15
16    Lmod = (D*deltat/deltax^2).*Lmtx;
17
18    yMtx = zeros(n+1,1);
19    for k = 1:n+1
20        if k == 1
21            yMtx(k) = 0;
22        else
23            yMtx(k) = con;
24        end
25    end
26
27    yMtx
28
29    %explicit Euler-módszer
30    z = zeros(n+1,1);
31    w = zeros(n+1,1);
32    for i = 1:(T/deltat)+1
33        if i == 1
34            yMtx = yMtx;
35        else
36            z = yMtx(:,i-1);
37            w = z+Lmod*z;
38            w(1) = 0;
39            yMtx = horzcat(yMtx,w);
40        end
41    end
42
43    %implicit Euler-módszer

```

```

44 %z2 = zeros(n+1,1);
45 %w2 = zeros(n+1,1);
46 %for i = 1:(T/deltat)+1
47 %     if i == 1
48 %         yMtx2 = yMtx2;
49 %     else
50 %         z2 = yMtx2(:, i-1);
51 %         w2 = (inv(eye(n+1)-Lmod))*z2;
52 %         w2(1) = 0;
53 %         yMtx2 = horzcat(yMtx2, w2);
54 %     end
55 %end
56
57 %Crank-Nicolson-modszer
58 %z3 = zeros(n+1,1);
59 %w3 = zeros(n+1,1);
60 %for i = 1:(T/deltat)+1
61 %     if i == 1
62 %         yMtx3 = yMtx3;
63 %     else
64 %         z3 = yMtx3(:, i-1);
65 %         w3 = ((inv(eye(n+1)-(0.5*Lmod)))*z3) + ((inv(eye(n+1)-(0.5*Lmod)))*(0.5*(
Lmod*z3)))
66 %         w3(1) = 0;
67 %         yMtx3 = horzcat(yMtx3, w3);
68 %     end
69 %end
70
71 %RK2
72 %z5 = zeros(n+1,1);
73 %w5 = zeros(n+1,1);
74 %for i = 1:(T/deltat)+1
75 %     if i == 1
76 %         yMtx5 = yMtx5;
77 %     else
78 %         z5 = yMtx5(:, i-1);
79 %         w5 = z5+(Lmod*z5)+(0.5*(Lmod^2)*z5);
80 %         w5(1) = 0;
81 %         yMtx5 = horzcat(yMtx5, w5);
82 %     end
83 %end
84
85 %RK3
86 %z6 = zeros(n+1,1);
87 %w6 = zeros(n+1,1);
88 %for i = 1:(T/deltat)+1
89 %     if i == 1
90 %         yMtx6 = yMtx6;
91 %     else
92 %         z6 = yMtx6(:, i-1);
93 %         w6 = z6+(Lmod*z6)+(0.5*(Lmod^2)*z6)+((1/6)*(Lmod^3)*z6);
94 %         w6(1) = 0;
95 %         yMtx6 = horzcat(yMtx6, w6);
96 %     end
97 %end
98
99 %z7 = zeros(n+1,1);
100 %w7 = zeros(n+1,1);
101 %for i = 1:(T/deltat)+1
102 %     if i == 1
103 %         yMtx7 = yMtx7;
104 %     else
105 %         z7 = yMtx7(:, i-1);

```

```

106 %      w7 = z7+(Lmod*z7)+(0.5*(Lmod^2)*z7)+((1/6)*(Lmod^3)*z7)+((1/24)*(Lmod^4)*
107 z7);
108 %      w7(1) = 0;
109 %      yMtx7 = horzcat(yMtx7,w7);
110 % end
111 %end
112
113 Tmtx = zeros(n+1,(T/deltat)+1);
114 for i = 1:n+1
115     for j = 1:(T/deltat)+1
116         Tmtx(i,j) = deltat*(j-1);
117     end
118 end
119 Tmtx
120
121 Xmtx = zeros(n+1,(T/deltat)+1);
122 for i = 1:n+1
123     for j = 1:(T/deltat)+1
124         Xmtx(i,:) = deltax*(i-1);
125     end
126 end
127 Xmtx
128
129 mesh(Tmtx,Xmtx,yMtx)
130 %mesh(Tmtx,Xmtx,yMtx2) %implicit Euler-modszer
131 %mesh(Tmtx,Xmtx,yMtx3) %Crank-Nicolson-modszer
132 %mesh(Tmtx,Xmtx,yMtx5) %RK2
133 %mesh(Tmtx,Xmtx,yMtx6) %RK3
134 %mesh(Tmtx,Xmtx,yMtx7) %RK4
135 xlabel('t');
136 ylabel('x');
137 zlabel('y');
138 grid on;
139 end

```

```

1 function [Tmtx,Xmtx] = adamsBashforth(n,D,deltat,deltax,con,T)
2
3 Lmtx = zeros(n+1,n+1);
4 for i = 1:n+1
5     for j = 1:n+1
6         if ((i == 1 && j == 1) || (i == n+1 && j == n+1))
7             Lmtx(i,j) = -1;
8         elseif i==j
9             Lmtx(i,j) = -2;
10        elseif i == j-1 || i == j+1
11            Lmtx(i,j) = 1;
12        end
13    end
14 end
15
16 Lmod = (D*deltat/deltax^2).*Lmtx;
17
18 yMtx8 = zeros(n+1,1);
19 for k = 1:n+1
20     if k == 1
21         yMtx8(k) = 0;
22     else
23         yMtx8(k) = con;
24     end
25 end
26
27 %ketlepeses
28 z8 = zeros(n+1,1);

```

```

29 w8 = zeros(n+1,1);
30 for i = 1:(T/deltat)+1
31     if i == 1
32         yMtx8 = yMtx8;
33     elseif i==2
34         z8 = yMtx8(:, i-1);
35         w8 = ((inv(eye(n+1)-(0.5*Lmod)))*z8) + ((inv(eye(n+1)-(0.5*Lmod)))*(0.5*(
Lmod*z8)));
36         w8(1) = 0;
37         yMtx8 = horzcat(yMtx8,w8);
38     else
39         z8a = yMtx8(:, i-1);
40         z8b = yMtx8(:, i-2);
41         w8b = ((inv(eye(n+1)-(2/3)*Lmod))*(4/3)*z8a) - (inv(eye(n+1)-(2/3)*Lmod)
*(1/3)*z8b);
42         w8b(1) = 0;
43         yMtx8 = horzcat(yMtx8,w8b);
44     end
45 end
46
47 %haromlepeses
48 z8 = zeros(n+1,1);
49 w8 = zeros(n+1,1);
50 for i = 1:(T/deltat)+1
51     if i == 1
52         yMtx8 = yMtx8;
53     elseif i==2 || i==3
54         z8 = yMtx8(:, i-1);
55         w8 = ((inv(eye(n+1)-(0.5*Lmod)))*z8) + ((inv(eye(n+1)-(0.5*Lmod)))*(0.5*(
Lmod*z8)));
56         w8(1) = 0;
57         yMtx8 = horzcat(yMtx8,w8);
58     else
59         z8a = yMtx8(:, i-1);
60         z8b = yMtx8(:, i-2);
61         z8c = yMtx8(:, i-3);
62         w8b = ((eye(n+1)+(23/12)*Lmod)*z8a) - (16/12)*Lmod*z8b + (5/12)*Lmod*z8c;
63         w8b(1) = 0;
64         yMtx8 = horzcat(yMtx8,w8b);
65     end
66 end
67
68 %negylepeses
69 z8 = zeros(n+1,1);
70 w8 = zeros(n+1,1);
71 for i = 1:(T/deltat)+1
72     if i == 1
73         yMtx8 = yMtx8;
74     elseif i==2 || i==3 || i==4
75         z8 = yMtx8(:, i-1);
76         w8 = ((inv(eye(n+1)-(0.5*Lmod)))*z8) + ((inv(eye(n+1)-(0.5*Lmod)))*(0.5*(
Lmod*z8)));
77         w8(1) = 0;
78         yMtx8 = horzcat(yMtx8,w8);
79     else
80         z8a = yMtx8(:, i-1);
81         z8b = yMtx8(:, i-2);
82         z8c = yMtx8(:, i-3);
83         z8d = yMtx8(:, i-4);
84         w8b = ((eye(n+1)+(55/24)*Lmod)*z8a) - (59/24)*Lmod*z8b + (37/24)*Lmod*z8c
-(9/24)*Lmod*z8d;
85         w8b(1) = 0;
86         yMtx8 = horzcat(yMtx8,w8b);

```

```

87     end
88 end
89
90 %otlepesees
91 z8 = zeros(n+1,1);
92 w8 = zeros(n+1,1);
93 for i = 1:(T/deltat)+1
94     if i == 1
95         yMtx8 = yMtx8;
96         elseif i==2 || i==3 || i==4 || i==5
97             z8 = yMtx8(:,i-1);
98             w8 = ((inv(eye(n+1)-(0.5*Lmod)))*z8) + ((inv(eye(n+1)-(0.5*Lmod)))*(0.5*(
Lmod*z8)));
99             w8(1) = 0;
100            yMtx8 = horzcat(yMtx8,w8);
101        else
102            z8a = yMtx8(:,i-1);
103            z8b = yMtx8(:,i-2);
104            z8c = yMtx8(:,i-3);
105            z8d = yMtx8(:,i-4);
106            z8e = yMtx8(:,i-5);
107            w8b = ((eye(n+1)+(1901/720)*Lmod)*z8a) - (2774/720)*Lmod*z8b + (2616/720)*
Lmod*z8c - (1272/720)*Lmod*z8d + (251/720)*Lmod*z8e;
108            w8b(1) = 0;
109            yMtx8 = horzcat(yMtx8,w8b);
110        end
111    end
112
113
114    Tmtx = zeros(n+1,(T/deltat)+1);
115    for i = 1:n+1
116        for j = 1:(T/deltat)+1
117            Tmtx(:,j) = deltat*(j-1);
118        end
119    end
120
121    Xmtx = zeros(n+1,(T/deltat)+1);
122    for i = 1:n+1
123        for j = 1:(T/deltat)+1
124            Xmtx(i,:) = deltax*(i-1);
125        end
126    end
127
128    surf(Tmtx,Xmtx,yMtx8)
129    xlabel('t');
130    ylabel('x');
131    zlabel('y');
132 end

```

```

1 function [] = adamsMoulton(n,D,deltat,deltax,con,T)
2
3     Lmtx = zeros(n+1,n+1);
4     for i = 1:n+1
5         for j = 1:n+1
6             if ((i == 1 && j== 1) || (i == n+1 && j == n+1))
7                 Lmtx(i,j) = -1;
8             elseif i==j
9                 Lmtx(i,j) = -2;
10            elseif i == j-1 || i == j+1
11                Lmtx(i,j) = 1;
12            end
13        end
14    end
15

```



```

16 Lmod = (D*deltat/deltax^2).*Lmtx;
17
18 yMtx8 = zeros(n+1,1);
19 for k = 1:n+1
20     if k == 1
21         yMtx8(k) = 0;
22     else
23         yMtx8(k) = con;
24     end
25 end
26
27 %ketlepeses
28 z8 = zeros(n+1,1);
29 w8 = zeros(n+1,1);
30 for i = 1:(T/deltat)+1
31     if i == 1
32         yMtx8 = yMtx8;
33     elseif i==2
34         z8 = yMtx8(:, i-1);
35         w8 = ((inv(eye(n+1)-(0.5*Lmod)))*z8) + ((inv(eye(n+1)-(0.5*Lmod)))*(0.5*(
Lmod*z8)));
36         w8(1) = 0;
37         yMtx8 = horzcat(yMtx8,w8);
38     else
39         z8a = yMtx8(:, i-1);
40         z8b = yMtx8(:, i-2);
41         w8b = ((inv(eye(n+1)-(5/12)*Lmod))*(eye(n+1)+(8/12)*Lmod))*z8a - ((inv(eye(
n+1)-(5/12)*Lmod))*(1/12)*Lmod))*z8b;
42         w8b(1) = 0;
43         yMtx8 = horzcat(yMtx8,w8b);
44     end
45 end
46
47 %haromlepeses
48 z8 = zeros(n+1,1);
49 w8 = zeros(n+1,1);
50 for i = 1:(T/deltat)+1
51     if i == 1
52         yMtx8 = yMtx8;
53     elseif i==2 || i==3
54         z8 = yMtx8(:, i-1);
55         w8 = ((inv(eye(n+1)-(0.5*Lmod)))*z8) + ((inv(eye(n+1)-(0.5*Lmod)))*(0.5*(
Lmod*z8)));
56         w8(1) = 0;
57         yMtx8 = horzcat(yMtx8,w8);
58     else
59         z8a = yMtx8(:, i-1);
60         z8b = yMtx8(:, i-2);
61         z8c = yMtx8(:, i-3);
62         w8b = (inv(eye(n+1)-(9/24)*Lmod))*(eye(n+1)+(19/24)*Lmod)*z8a - (inv(eye(n
+1)-(9/24)*Lmod))*(5/24)*Lmod*z8b + (inv(eye(n+1)-(9/24)*Lmod))*(1/24)*Lmod*z8c;
63         w8b(1) = 0;
64         yMtx8 = horzcat(yMtx8,w8b);
65     end
66 end
67
68 %negylepeses
69 z8 = zeros(n+1,1);
70 w8 = zeros(n+1,1);
71 for i = 1:(T/deltat)+1
72     if i == 1
73         yMtx8 = yMtx8;
74     elseif i==2 || i==3 || i==4

```

```

75     z8 = yMtx8(:, i-1);
76     w8 = ((inv(eye(n+1)-(0.5*Lmod)))*z8) + ((inv(eye(n+1)-(0.5*Lmod)))*(0.5*(
Lmod*z8)));
77     w8(1) = 0;
78     yMtx8 = horzcat(yMtx8, w8);
79     else
80         z8a = yMtx8(:, i-1);
81         z8b = yMtx8(:, i-2);
82         z8c = yMtx8(:, i-3);
83         z8d = yMtx8(:, i-4);
84         w8b = (inv(eye(n+1)-(251/720)*Lmod))*(eye(n+1)+(646/720)*Lmod)*z8a - (inv(
eye(n+1)-(251/720)*Lmod))*(264/720)*Lmod*z8b + (inv(eye(n+1)-(251/720)*Lmod))
*(106/720)*Lmod*z8c - (inv(eye(n+1)-(251/720)*Lmod))*(19/720)*Lmod*z8d;
85         w8b(1) = 0;
86         yMtx8 = horzcat(yMtx8, w8b);
87     end
88 end
89
90 %otlepeses
91 z8 = zeros(n+1,1);
92 w8 = zeros(n+1,1);
93 for i = 1:(T/deltat)+1
94     if i == 1
95         yMtx8 = yMtx8;
96     elseif i==2 || i==3 || i==4 || i==5
97         z8 = yMtx8(:, i-1);
98         w8 = ((inv(eye(n+1)-(0.5*Lmod)))*z8) + ((inv(eye(n+1)-(0.5*Lmod)))*(0.5*(
Lmod*z8)));
99         w8(1) = 0;
100        yMtx8 = horzcat(yMtx8, w8);
101    else
102        z8a = yMtx8(:, i-1);
103        z8b = yMtx8(:, i-2);
104        z8c = yMtx8(:, i-3);
105        z8d = yMtx8(:, i-4);
106        z8e = yMtx8(:, i-5);
107        w8b = (inv(eye(n+1)-(475/1440)*Lmod))*(eye(n+1)+(1427/1440)*Lmod)*z8a - (
inv(eye(n+1)-(475/1440)*Lmod))*(798/1440)*Lmod*z8b + (inv(eye(n+1)-(475/1440)*Lmod)
)*(482/1440)*Lmod*z8c - (inv(eye(n+1)-(475/1440)*Lmod))*(173/1440)*Lmod*z8d + (inv(
eye(n+1)-(475/1440)*Lmod))*(27/1440)*Lmod*z8e;
108        w8b(1) = 0;
109        yMtx8 = horzcat(yMtx8, w8b);
110    end
111 end
112
113 Tmtx = zeros(n+1,(T/deltat)+1);
114 for i = 1:n+1
115     for j = 1:(T/deltat)+1
116         Tmtx(:, j) = deltat*(j-1);
117     end
118 end
119
120 Xmtx = zeros(n+1,(T/deltat)+1);
121 for i = 1:n+1
122     for j = 1:(T/deltat)+1
123         Xmtx(i, :) = deltax*(i-1);
124     end
125 end
126
127 surf(Tmtx, Xmtx, yMtx8)
128 xlabel('t');
129 ylabel('x');
130 zlabel('y');

```

```

131 end

1 function [] = adamsBashforthmod(n,D,deltat ,deltax ,con ,T)
2
3 Accurate = zeros(n+1,(T/deltat)+1);
4 for i = 1:(n+1)
5     for j = 1:(T/deltat)+1
6         if j== 1
7             Accurate(i,:) = 1;
8         else
9             auxVar = deltax*(i-1)/(sqrt(4*D*(j-1)*deltat));
10            Accurate(i,j) = erf(auxVar);
11        end
12    end
13 end
14 Accurate2 = con*Accurate;
15
16 Lmtx = zeros(n+1,n+1);
17 for i = 1:n+1
18     for j = 1:n+1
19         if ((i == 1 && j== 1) || (i == n+1 && j == n+1))
20             Lmtx(i,j) = -1;
21         elseif i==j
22             Lmtx(i,j) = -2;
23         elseif i == j-1 || i == j+1
24             Lmtx(i,j) = 1;
25         end
26     end
27 end
28
29 Lmod = (D*deltat/deltax^2).*Lmtx;
30
31 yMtx8 = zeros(n+1,1);
32 for k = 1:n+1
33     if k == 1
34         yMtx8(k) = 0;
35     else
36         yMtx8(k) = con;
37     end
38 end
39
40 %ketlepeses
41 z8 = zeros(n+1,1);
42 w8 = zeros(n+1,1);
43 for i = 1:(T/deltat)+1
44     if i == 1
45         yMtx8 = yMtx8;
46     elseif i==2
47         z8 = yMtx8(:,i-1);
48         w8 = ((inv(eye(n+1)-(0.5*Lmod)))*z8) + ((inv(eye(n+1)-(0.5*Lmod)))*(0.5*(
49 Lmod*z8)));
50         w8(1) = 0;
51         yMtx8 = horzcat(yMtx8,w8);
52     else
53         z8a = Accurate2(:,i-1);
54         z8b = Accurate2(:,i-2);
55         w8b = ((eye(n+1)+(3/2)*Lmod)*z8a) - (1/2)*Lmod*z8b;
56         w8b(1) = 0;
57         yMtx8 = horzcat(yMtx8,w8b);
58     end
59 end
60
61 %haromlepeses
62 z8 = zeros(n+1,1);

```

```

62 w8 = zeros(n+1,1);
63 for i = 1:(T/deltat)+1
64     if i == 1
65         yMtx8 = yMtx8;
66     elseif i==2 || i==3
67         z8 = yMtx8(:, i-1);
68         w8 = ((inv(eye(n+1)-(0.5*Lmod)))*z8) + ((inv(eye(n+1)-(0.5*Lmod)))*(0.5*(
Lmod*z8)));
69         w8(1) = 0;
70         yMtx8 = horzcat(yMtx8,w8);
71     else
72         z8a = Accurate2(:, i-1);
73         z8b = Accurate2(:, i-2);
74         z8c = Accurate2(:, i-3);
75         w8b = ((eye(n+1)+(23/12)*Lmod)*z8a) - (16/12)*Lmod*z8b + (5/12)*Lmod*z8c;
76         w8b(1) = 0;
77         yMtx8 = horzcat(yMtx8,w8b);
78     end
79 end
80
81 %negylepeses
82 z8 = zeros(n+1,1);
83 w8 = zeros(n+1,1);
84 for i = 1:(T/deltat)+1
85     if i == 1
86         yMtx8 = yMtx8;
87     elseif i==2 || i==3 || i==4
88         z8 = yMtx8(:, i-1);
89         w8 = ((inv(eye(n+1)-(0.5*Lmod)))*z8) + ((inv(eye(n+1)-(0.5*Lmod)))*(0.5*(
Lmod*z8)));
90         w8(1) = 0;
91         yMtx8 = horzcat(yMtx8,w8);
92     else
93         z8a = Accurate2(:, i-1);
94         z8b = Accurate2(:, i-2);
95         z8c = Accurate2(:, i-3);
96         z8d = Accurate2(:, i-4);
97         w8b = ((eye(n+1)+(55/24)*Lmod)*z8a) - (59/24)*Lmod*z8b + (37/24)*Lmod*z8c
- (9/24)*Lmod*z8d;
98         w8b(1) = 0;
99         yMtx8 = horzcat(yMtx8,w8b);
100    end
101 end
102
103 %otlepeses
104 z8 = zeros(n+1,1);
105 w8 = zeros(n+1,1);
106 for i = 1:(T/deltat)+1
107     if i == 1
108         yMtx8 = yMtx8;
109     elseif i==2 || i==3 || i==4 || i==5
110         z8 = yMtx8(:, i-1);
111         w8 = ((inv(eye(n+1)-(0.5*Lmod)))*z8) + ((inv(eye(n+1)-(0.5*Lmod)))*(0.5*(
Lmod*z8)));
112         w8(1) = 0;
113         yMtx8 = horzcat(yMtx8,w8);
114     else
115         z8a = Accurate2(:, i-1);
116         z8b = Accurate2(:, i-2);
117         z8c = Accurate2(:, i-3);
118         z8d = Accurate2(:, i-4);
119         z8e = Accurate2(:, i-5);
120         w8b = ((eye(n+1)+(1901/720)*Lmod)*z8a) - (2774/720)*Lmod*z8b + (2616/720)*

```

```

121     Lmod*z8c = (1272/720)*Lmod*z8d + (251/720)*Lmod*z8e;
122     w8b(1) = 0;
123     yMtx8 = horzcat(yMtx8,w8b);
124 end
125
126
127 Tmtx = zeros(n+1,(T/deltat)+1);
128 for i = 1:n+1
129     for j = 1:(T/deltat)+1
130         Tmtx(:,j) = deltax*(j-1);
131     end
132 end
133
134 Xmtx = zeros(n+1,(T/deltat)+1);
135 for i = 1:n+1
136     for j = 1:(T/deltat)+1
137         Xmtx(i,:) = deltax*(i-1);
138     end
139 end
140
141 surf(Tmtx,Xmtx,yMtx8)
142 xlabel('t');
143 ylabel('x');
144 zlabel('y');
145 end

```

```

1 function [] = adamsMoultonmod(n,D,deltat,deltax,con,T)
2
3     Accurate = zeros(n+1,(T/deltat)+1);
4     for i = 1:(n+1)
5         for j = 1:(T/deltat)+1
6             if j== 1
7                 Accurate(i,:) = 1;
8             else
9                 auxVar = deltax*(i-1)/(sqrt(4*D*(j-1)*deltat));
10                Accurate(i,j) = erf(auxVar);
11            end
12        end
13    end
14    Accurate2 = con*Accurate;
15
16    Lmtx = zeros(n+1,n+1);
17    for i = 1:n+1
18        for j = 1:n+1
19            if ((i == 1 && j== 1) || (i == n+1 && j == n+1))
20                Lmtx(i,j) = -1;
21            elseif i==j
22                Lmtx(i,j) = -2;
23            elseif i == j-1 || i == j+1
24                Lmtx(i,j) = 1;
25            end
26        end
27    end
28
29    Lmod = (D*deltat/deltax^2).*Lmtx;
30
31    yMtx8 = zeros(n+1,1);
32    for k = 1:n+1
33        if k == 1
34            yMtx8(k) = 0;
35        else
36            yMtx8(k) = con;
37        end

```

```

38 end
39
40 %ketlepeses
41 z8 = zeros(n+1,1);
42 w8 = zeros(n+1,1);
43 for i = 1:(T/deltat)+1
44     if i == 1
45         yMtx8 = yMtx8;
46     elseif i==2
47         z8 = yMtx8(:, i-1);
48         w8 = ((inv(eye(n+1)-(0.5*Lmod)))*z8) + ((inv(eye(n+1)-(0.5*Lmod)))*(0.5*(
Lmod*z8)));
49         w8(1) = 0;
50         yMtx8 = horzcat(yMtx8,w8);
51     else
52         z8a = Accurate2(:, i-1);
53         z8b = Accurate2(:, i-2);
54         w8b = ((inv(eye(n+1)-(5/12)*Lmod))*(eye(n+1)+(8/12)*Lmod))*z8a - ((inv(eye(
n+1)-(5/12)*Lmod))*(1/12)*Lmod))*z8b;
55         w8b(1) = 0;
56         yMtx8 = horzcat(yMtx8,w8b);
57     end
58 end
59
60 %haromlepeses
61 z8 = zeros(n+1,1);
62 w8 = zeros(n+1,1);
63 for i = 1:(T/deltat)+1
64     if i == 1
65         yMtx8 = yMtx8;
66     elseif i==2 || i==3
67         z8 = yMtx8(:, i-1);
68         w8 = ((inv(eye(n+1)-(0.5*Lmod)))*z8) + ((inv(eye(n+1)-(0.5*Lmod)))*(0.5*(
Lmod*z8)));
69         w8(1) = 0;
70         yMtx8 = horzcat(yMtx8,w8);
71     else
72         z8a = Accurate2(:, i-1);
73         z8b = Accurate2(:, i-2);
74         z8c = Accurate2(:, i-3);
75         w8b = (inv(eye(n+1)-(9/24)*Lmod))*(eye(n+1)+(19/24)*Lmod)*z8a - (inv(eye(n
+1)-(9/24)*Lmod))*(5/24)*Lmod*z8b + (inv(eye(n+1)-(9/24)*Lmod))*(1/24)*Lmod*z8c;
76         w8b(1) = 0;
77         yMtx8 = horzcat(yMtx8,w8b);
78     end
79 end
80
81 %negylepeses
82 z8 = zeros(n+1,1);
83 w8 = zeros(n+1,1);
84 for i = 1:(T/deltat)+1
85     if i == 1
86         yMtx8 = yMtx8;
87     elseif i==2 || i==3 || i==4
88         z8 = yMtx8(:, i-1);
89         w8 = ((inv(eye(n+1)-(0.5*Lmod)))*z8) + ((inv(eye(n+1)-(0.5*Lmod)))*(0.5*(
Lmod*z8)));
90         w8(1) = 0;
91         yMtx8 = horzcat(yMtx8,w8);
92     else
93         z8a = Accurate2(:, i-1);
94         z8b = Accurate2(:, i-2);
95         z8c = Accurate2(:, i-3);

```

```

96         z8d = Accurate2(:,i-4);
97         w8b = (inv(eye(n+1)-(251/720)*Lmod))*(eye(n+1)+(646/720)*Lmod)*z8a - (inv(
eye(n+1)-(251/720)*Lmod))*(264/720)*Lmod*z8b + (inv(eye(n+1)-(251/720)*Lmod))
*(106/720)*Lmod*z8c - (inv(eye(n+1)-(251/720)*Lmod))*(19/720)*Lmod*z8d;
98         w8b(1) = 0;
99         yMtx8 = horzcat(yMtx8,w8b);
100     end
101 end
102
103 %otlepeses
104 z8 = zeros(n+1,1);
105 w8 = zeros(n+1,1);
106 for i = 1:(T/deltat)+1
107     if i == 1
108         yMtx8 = yMtx8;
109     elseif i==2 || i==3 || i==4 || i==5
110         z8 = yMtx8(:,i-1);
111         w8 = ((inv(eye(n+1)-(0.5*Lmod)))*z8) + ((inv(eye(n+1)-(0.5*Lmod)))*(0.5*(
Lmod*z8)));
112         w8(1) = 0;
113         yMtx8 = horzcat(yMtx8,w8);
114     else
115         z8a = Accurate2(:,i-1);
116         z8b = Accurate2(:,i-2);
117         z8c = Accurate2(:,i-3);
118         z8d = Accurate2(:,i-4);
119         z8e = Accurate2(:,i-5);
120         w8b = (inv(eye(n+1)-(475/1440)*Lmod))*(eye(n+1)+(1427/1440)*Lmod)*z8 - (inv
(eye(n+1)-(475/1440)*Lmod))*(798/1440)*Lmod*z8b + (inv(eye(n+1)-(475/1440)*Lmod))
*(482/1440)*Lmod*z8c - (inv(eye(n+1)-(475/1440)*Lmod))*(173/1440)*Lmod*z8d + (inv(
eye(n+1)-(475/1440)*Lmod))*(27/1440)*Lmod*z8e;
121         w8b(1) = 0;
122         yMtx8 = horzcat(yMtx8,w8b);
123     end
124 end
125
126 Tmtx = zeros(n+1,(T/deltat)+1);
127 for i = 1:n+1
128     for j = 1:(T/deltat)+1
129         Tmtx(:,j) = deltax*(j-1);
130     end
131 end
132
133 Xmtx = zeros(n+1,(T/deltat)+1);
134 for i = 1:n+1
135     for j = 1:(T/deltat)+1
136         Xmtx(i,:) = deltax*(i-1);
137     end
138 end
139
140 surf(Tmtx,Xmtx,yMtx8)
141 xlabel('t');
142 ylabel('x');
143 zlabel('y');
144 end

```

```

1 %hiba1
2 T = [0.001, 0.0014, 0.0016, 0.002, 0.0025, 0.0028, 0.0035, 0.004, 0.005, 0.0056, 0.007,
0.008, 0.01, 0.014, 0.02, 0.025, 0.028, 0.04, 0.05, 0.056, 0.1, 0.2, 0.25, 0.35,
0.5, 0.7, 1, 1.75, 3.5];
3 expNorm = [4.034510867083395e-09, 4.030678261585209e-09, 4.028778828679855e-09,
4.025013784196629e-09, 4.020371060700723e-09, 4.017619433618730e-09,
4.011298506095140e-09, 4.006869169732769e-09, 3.998225595459733e-09,
3.993177756294022e-09, 3.981805475447188e-09, 3.974032614663371e-09,

```

```

3.959370131719035e-09, 3.933624273956815e-09, 3.904128339383817e-09,
3.888074366342882e-09, 3.882211543608965e-09, 3.887267719025153e-09,
3.926154772006269e-09, 3.964275288506906e-09, 4.543725951723191e-09,
7.058871775369539e-09, 8.595401112403599e-09, 1.188131332448279e-08,
2.486568219480798e-08, 1.006463368944193e-05, 9.074568580476495e-05,
9.124025842006176e-05, 2.169546645546567e-05];
4 impNorm = [4.075855286007086e-09, 4.097078567109571e-09, 4.109497522286074e-09,
4.137894385693718e-09, 4.179932007448862e-09, 4.208558807074908e-09,
4.284941027300177e-09, 4.347398301235917e-09, 4.490813799925003e-09,
4.587912329168243e-09, 4.843384553124707e-09, 5.047891762991111e-09,
5.502737241852109e-09, 6.548479296064080e-09, 8.322408240983589e-09,
9.902442067628818e-09, 1.087622058029732e-08, 1.487357172401383e-08,
1.826509141480092e-08, 2.031084224845416e-08, 3.532692177193719e-08,
6.873273747723829e-08, 8.494803705377904e-08, 1.163664953306375e-07,
1.609902515874991e-07, 2.160023511285649e-07, 2.896897563079865e-07,
4.360026107420190e-07, 6.461550155816964e-07];
5 cnNorm = [4.050597764338610e-09, 4.054940046703093e-09, 4.057499267355146e-09,
4.063391253057249e-09, 4.072200307938695e-09, 4.078251750585098e-09,
4.094588451673216e-09, 4.108140467383563e-09, 4.139880538077867e-09,
4.161842471357647e-09, 4.221366586036696e-09, 4.270760762395597e-09,
4.385857124456308e-09, 4.675161418439969e-09, 5.230368063240859e-09,
5.776835695620229e-09, 6.132194995436098e-09, 7.693033396612167e-09,
9.100426300394910e-09, 9.972447481121390e-09, 1.663800449048359e-08,
3.209852224517187e-08, 3.976715194331207e-08, 5.488234433622909e-08,
7.694678721061268e-08, 1.052106027387148e-07, 1.451980599954727e-07,
2.339493686944197e-07, 3.975865087459658e-07];
6 rk2Norm = [4.050615227591415e-09, 4.054978513270531e-09, 4.057552272360510e-09,
4.063482676941248e-09, 4.072359872518430e-09, 4.078464415335975e-09,
4.094965964762919e-09, 4.108675267629288e-09, 4.140844409735744e-09,
4.163146160482256e-09, 4.223738662600086e-09, 4.274159458193999e-09,
4.392054639876871e-09, 4.690352039965734e-09, 5.268547741609846e-09,
5.843293795481965e-09, 6.219606744558711e-09, 7.893688034674281e-09,
9.431007651565133e-09, 1.039653075706368e-08, 1.812044151849261e-08,
3.891410326179201e-08, 5.128389294910502e-08, 8.263114165177621e-08,
2.406335482328160e-07, 8.436964676125266e-05, 0.003717774442025, 0.007870552563827,
8.174243892454038e-04];
7 rk3Norm = [4.050603578209768e-09, 4.054952839681883e-09, 4.057516890206228e-09,
4.063421624104782e-09, 4.072253263250136e-09, 4.078322283807793e-09,
4.094713472646931e-09, 4.108317387358965e-09, 4.140198704165665e-09,
4.162272236786939e-09, 4.222146112028504e-09, 4.271875163393757e-09,
4.387880141138599e-09, 4.680075901759322e-09, 5.242556947226716e-09,
5.797819250288244e-09, 6.159611160300456e-09, 7.754279344784795e-09,
9.198982042986675e-09, 1.009705051251197e-08, 1.702509030380966e-08,
3.333809299176080e-08, 4.139606757353032e-08, 5.669648950949593e-08,
7.611753118451097e-08, 1.424870499637236e-06, 0.005756889841403, 0.116321661191085,
0.012694082994760];
8 rk4Norm = [4.050603581672840e-09, 4.054952852667842e-09, 4.057516910492153e-09,
4.063421671529403e-09, 4.072253369682145e-09, 4.078322445484235e-09,
4.094713842377246e-09, 4.108317996755789e-09, 4.140200115102617e-09,
4.162274402427128e-09, 4.222151156603743e-09, 4.271883533278824e-09,
4.387899611111106e-09, 4.680144562441530e-09, 5.242809994721903e-09,
5.798378607857379e-09, 6.160441864814956e-09, 7.757077655626837e-09,
9.204853704947106e-09, 1.010557429545796e-08, 1.708185517373113e-08,
3.391725048352689e-08, 4.265449512313523e-08, 6.087838280750746e-08,
9.176538912287090e-08, 2.165170665342617e-07, 0.002633576184367, 0.654145950317708,
0.118338695987661];
9 plot(T, expNorm, T, impNorm, T, cnNorm, T, rk2Norm, T, rk3Norm, T, rk4Norm, [.5 .5], [1E
-10 0.7], 'r—')
10 xlabel('\Deltat (idobeli lepeskozok)')
11 ylabel('Hiba (2-es normaban)')
12 legend('explicit Euler-modszer', 'implicit Euler-modszer', 'Crank-Nicolson-modszer', '
Masodrendu Runge-Kutta', 'Harmadrendu Runge-Kutta', 'Negyedrendu Runge-Kutta', '\
Deltat =0,5')

```



```

13
14 %hib2
15 T = [0.001, 0.0014, 0.0016, 0.002, 0.0025, 0.0028, 0.0035, 0.004, 0.005, 0.0056, 0.007,
      0.008, 0.01, 0.014, 0.02, 0.025, 0.028, 0.04, 0.05, 0.056, 0.1, 0.2, 0.25, 0.35,
      0.5, 0.7, 1, 1.75, 3.5];
16 expNorm = [4.034510867083395e-09, 4.030678261585209e-09, 4.028778828679855e-09,
      4.025013784196629e-09, 4.020371060700723e-09, 4.017619433618730e-09,
      4.011298506095140e-09, 4.006869169732769e-09, 3.998225595459733e-09,
      3.993177756294022e-09, 3.981805475447188e-09, 3.974032614663371e-09,
      3.959370131719035e-09, 3.933624273956815e-09, 3.904128339383817e-09,
      3.888074366342882e-09, 3.882211543608965e-09, 3.887267719025153e-09,
      3.926154772006269e-09, 3.964275288506906e-09, 4.543725951723191e-09,
      7.058871775369539e-09, 8.595401112403599e-09, 1.188131332448279e-08,
      2.486568219480798e-08, 1.006463368944193e-05, 9.074568580476495e-05,
      9.124025842006176e-05, 2.169546645546567e-05];
17 ab2 = [4.044278818522515e-09, 4.044269606214732e-09, 4.044263853098642e-09,
      4.044250054338767e-09, 4.044228517296339e-09, 4.044213312712296e-09,
      4.044171194734711e-09, 4.044135433926583e-09, 4.044049776307304e-09,
      4.043989370607358e-09, 4.043822294287736e-09, 4.043680693523387e-09,
      4.043342362158573e-09, 4.042448735426625e-09, 4.040582655039819e-09,
      4.038567203872856e-09, 4.037166714153116e-09, 4.030248256980388e-09,
      4.023077164119215e-09, 4.018271946064367e-09, 3.978173720550604e-09,
      3.999710385031598e-09, 4.287745875081584e-09, 1.036948164465430e-05,
      5.165048074786812e-04, 9.702900468586559e-04, 3.833562036708574e-04,
      2.654190217698745e-05, 8.678412614669950e-07];
18 ab3 = [4.044260294616761e-09, 4.044233321678829e-09, 4.044216476093809e-09,
      4.044176076154487e-09, 4.044113020172227e-09, 4.044068503797253e-09,
      4.043945191546196e-09, 4.043840492519317e-09, 4.043589712031543e-09,
      4.043412862584198e-09, 4.042923714907758e-09, 4.042509146847544e-09,
      4.041518573030444e-09, 4.038901837168107e-09, 4.033435164958236e-09,
      4.027526348956222e-09, 4.023417216860598e-09, 4.003071389071200e-09,
      3.981883437383282e-09, 3.967617494891843e-09, 3.844867437559321e-09,
      6.653046102196717e-05, 0.003226986597659, 0.036865692052546, 0.029256460480917,
      0.005677584928598, 5.178584608841739e-04, 1.011162151772917e-05, 3.975865087459658e-07];
19 ab4 = [4.044246255157026e-09, 4.044205841461517e-09, 4.044180608535552e-09,
      4.044120111887757e-09, 4.044025732115350e-09, 4.043959128091350e-09,
      4.043774727604971e-09, 4.043618257246309e-09, 4.043243775100845e-09,
      4.042979917565129e-09, 4.042250949626049e-09, 4.041633974191866e-09,
      4.040162483556555e-09, 4.036290769017125e-09, 4.028261194899491e-09,
      4.019658888880670e-09, 4.013719588252878e-09, 3.984796917146707e-09,
      3.955519756673450e-09, 3.936308697628484e-09, 1.493616002403032e-05,
      0.816237438001284, 0.306261056547849, 0.526583000297258, 0.058519495124047,
      0.003168974231994, 1.117723315669500e-04, 8.543819028814144e-07, 3.975865087459658e-07];
20 ab5 = [4.773950563085380e-09, 4.773658942635327e-09, 4.773502313044361e-09,
      4.773167475418439e-09, 4.772708577714896e-09, 4.772411786490514e-09,
      4.771656945610842e-09, 4.771064562353567e-09, 4.769747542499862e-09,
      4.768873167818315e-09, 4.766589496152213e-09, 4.764751358424403e-09,
      4.760564177851005e-09, 4.750188025357924e-09, 4.729803474689461e-09,
      4.708615864760156e-09, 4.694162252664933e-09, 4.624323037659307e-09,
      9.261358685870293e-06, 0.004367021672125, 1.412273298436004e+04, 5.049636613922997e+03,
      5.947249943832701e+02, 12.411348427392715, 0.144768614168320,
      0.002168130803384, 3.173159333076087e-05, 2.339493686944197e-07, 3.975865087459658e-07];
21 plot(T, expNorm, T, ab2, T, ab3, T, ab4, T, ab5, [.5 .5], [1E-10 10^5], 'r—')
22 %loglog([.5 .5], [0.0000001 10], 'r')
23 xlabel('\Deltat (idobeli lepeskozok)')
24 ylabel('Hiba (2-es normaban)')
25 legend('explicit Euler-modszer', 'ketlepeses Adams-Bashforth', 'haromlepeses Adams-Bashforth', 'negylepeses Adams-Bashforth', 'otlepeses Adams-Bashforth', '\Deltat =0,5')

```

26

```

27 %hiba3
28 T = [0.001, 0.0014, 0.0016, 0.002, 0.0025, 0.0028, 0.0035, 0.004, 0.005, 0.0056, 0.007,
      0.008, 0.01, 0.014, 0.02, 0.025, 0.028, 0.04, 0.05, 0.056, 0.1, 0.2, 0.25, 0.35,
      0.5, 0.7, 1, 1.75, 3.5];
29 impNorm = [4.075855286007086e-09, 4.097078567109571e-09, 4.109497522286074e-09,
      4.137894385693718e-09, 4.179932007448862e-09, 4.208558807074908e-09,
      4.284941027300177e-09, 4.347398301235917e-09, 4.490813799925003e-09,
      4.587912329168243e-09, 4.843384553124707e-09, 5.047891762991111e-09,
      5.502737241852109e-09, 6.548479296064080e-09, 8.322408240983589e-09,
      9.902442067628818e-09, 1.087622058029732e-08, 1.487357172401383e-08,
      1.826509141480092e-08, 2.031084224845416e-08, 3.532692177193719e-08,
      6.873273747723829e-08, 8.494803705377904e-08, 1.163664953306375e-07,
      1.609902515874991e-07, 2.160023511285649e-07, 2.896897563079865e-07,
      4.360026107420190e-07, 6.461550155816964e-07];
30 cnNorm = [4.050597764338610e-09, 4.054940046703093e-09, 4.057499267355146e-09,
      4.063391253057249e-09, 4.072200307938695e-09, 4.078251750585098e-09,
      4.094588451673216e-09, 4.108140467383563e-09, 4.139880538077867e-09,
      4.161842471357647e-09, 4.221366586036696e-09, 4.270760762395597e-09,
      4.385857124456308e-09, 4.675161418439969e-09, 5.230368063240859e-09,
      5.776835695620229e-09, 6.132194995436098e-09, 7.693033396612167e-09,
      9.100426300394910e-09, 9.972447481121390e-09, 1.663800449048359e-08,
      3.209852224517187e-08, 3.976715194331207e-08, 5.488234433622909e-08,
      7.694678721061268e-08, 1.052106027387148e-07, 1.451980599954727e-07,
      2.339493686944197e-07, 3.975865087459658e-07];
31 am2Norm = [4.049093070756075e-09, 4.052278289578986e-09, 4.054140819129259e-09,
      4.058404429554144e-09, 4.064740617928776e-09, 4.069077171970617e-09,
      4.080747102619926e-09, 4.090404023861694e-09, 4.112985577289715e-09,
      4.128601268969063e-09, 4.170949412973548e-09, 4.206152721610392e-09,
      4.288485115835175e-09, 4.497542470494049e-09, 4.906809849263132e-09,
      5.318387262025086e-09, 5.589857939996501e-09, 6.808539434635110e-09,
      7.932663151350346e-09, 8.637186133965362e-09, 1.412867728395843e-08,
      2.715077471101033e-08, 3.368566697815354e-08, 4.667804800403984e-08,
      6.588727768549367e-08, 9.090266925260765e-08, 1.269127392318345e-07,
      2.177622505272534e-07, 3.961506589843989e-07];
32 am3Norm = [4.048404613615564e-09, 4.051072400752190e-09, 4.052624676838292e-09,
      4.056165178332541e-09, 4.061406297619687e-09, 4.064984614424622e-09,
      4.074593365766605e-09, 4.082530834860951e-09, 4.101068280727759e-09,
      4.113878890202487e-09, 4.148618176541075e-09, 4.177513354109448e-09,
      4.245200379454594e-09, 4.417908261538057e-09, 4.759517499682406e-09,
      5.107163365933599e-09, 5.338378861895454e-09, 6.390678455549991e-09,
      7.376222778113286e-09, 7.998985723758089e-09, 1.292830540815515e-08,
      2.484880661798671e-08, 3.089819797928877e-08, 4.302319402188309e-08,
      6.115396540867135e-08, 8.525813389845631e-08, 1.204444442980191e-07,
      2.218174259760812e-07, 3.975865087459658e-07];
33 am4Norm = [4.047989293163101e-09, 4.050349282843956e-09, 4.051717489495125e-09,
      4.054829822772694e-09, 4.059423836456194e-09, 4.062554647162623e-09,
      4.070948198133894e-09, 4.077872766413253e-09, 4.094029059102752e-09,
      4.105188236251438e-09, 4.135446244290365e-09, 4.160622788738352e-09,
      4.219660488046964e-09, 4.370792437744131e-09, 4.671848779786064e-09,
      4.980799390345887e-09, 5.187515828447040e-09, 6.138059536175462e-09,
      7.038950858087814e-09, 7.612043583205838e-09, 1.220901280098033e-08,
      2.353285757060577e-08, 2.934480769544520e-08, 4.109541596938592e-08,
      5.889245820741258e-08, 8.326322133052523e-08, 1.198497700611716e-07,
      2.167934861613539e-07, 3.975865087459658e-07];
34 am5Norm = [1.130537733832192e-06, 1.129434341060345e-06, 1.128883991896582e-06,
      1.127785975418943e-06, 1.126418457477087e-06, 1.125600599863830e-06,
      1.123699949787933e-06, 1.122348883276930e-06, 1.119662935404374e-06,
      1.118061624603516e-06, 1.114354738695324e-06, 1.111731905741815e-06,
      1.106547346376808e-06, 1.096414970130594e-06, 1.081777047512748e-06,
      1.070059346555966e-06, 1.063225813686331e-06, 1.037248184084929e-06,
      1.017093221085929e-06, 1.005580514783433e-06, 9.317813713813515e-07,
      8.077163634987858e-07, 7.590101762265984e-07, 6.773701583493978e-07,
      5.803032749951710e-07, 4.795818493985472e-07, 3.660067619823541e-07,

```

```
2.339493686944197e-07, 3.975865087459658e-07];  
35 plot(T, impNorm, T, cnNorm, T, am2Norm, T, am3Norm, T, am4Norm, T, am5Norm, [.5 .5], [1E  
-10 0.7], 'r—')  
36 xlabel('\Deltat (idobeli lepeskozok)')  
37 ylabel('Hiba (2-es normaban)')  
38 legend('implicit Euler-modszer', 'Crank-Nicolson-modszer', 'ketlepeses Adams-Moulton',  
'haromlepeses Adams-Moulton', 'negylepeses Adams-Moulton', 'otlepeses Adams-Moulton',  
, '\Deltat =0,5')
```

Irodalomjegyzék

- [1] Vesztergom Soma: *A short introduction to digital simulation in electrochemistry: simulating the Cottrell experiment in NI LabVIEW*
Journal of Electrochemical Science and Engineering, Vol.8 No.2 Svibanj (2018)
<https://hrcak.srce.hr/200268>
- [2] J. P. Gambino, Sh. A. Adderly, J. Knickerbocker, *Microelectronic Engineering*, 135, 73–106. (2015)
- [3] A. J. Bard, L. R. Faulkner *Electrochemical Methods: Fundamentals and Applications*, John Wiley & Sons, New York, (2005), pp. 785–807.
- [4] F. G. Cottrell - *Zeitschrift für physikalische Chemie*, 42 (1903) 385–431
- [5] A. J. Bard, Gy. Inzelt, F. Scholz (Eds) – *Electrochemical Dictionary*, Springer, Berlin (2008), p. 163
- [6] P. W. Atkins: *Fizikai kémia I.-III.*, Nemzeti Tankönyvkiadó (2002)
- [7] Inzelt György: *Az elektrokémia korszerű elmélete és módszerei*
- [8] J. Crank: *The Mathematics of Diffusion (Second Edition)*, Oxford (1975)
- [9] Benoit Cushman-Roisin: *Environmental Transport and Fate*, Dartmouth College (2012)
- [10] Guy Simpson: *Practical Finite Element Modeling in Earth Science Using Matlab* (2017)
- [11] Greg Fasshauer jegyzete
http://www.math.iit.edu/~fass/478578_Chapter_2.pdf
- [12] Faragó István, Horváth Róbert: *Numerikus módszerek* (2013)
<http://tankonyvtar.ttk.bme.hu/pdf/30.pdf>
- [13] Faragó István: *Numerikus modellezés és közönséges differenciálegyenletek numerikus megoldási módszerei* (2013.02.15.)
http://web.cs.elte.hu/~faragois/jegyzet_Szeged.pdf
- [14] Faragó István előadásai
- [15] Besenyei Ádám, Komornik Vilmos, Simon László: *Parciális differenciálegyenletek*, Typotex Kiadó (2013)
http://etananyag.ttk.elte.hu/Files/downloads/_Besenyei_Parc_diffegyenlet.pdf
- [16] Mezei István, Faragó István, Simon Péter: *Bevezetés az analízisbe*, Typotex Kiadó (2013)
- [17] Simon L. Péter: *Közönséges Differenciálegyenletek jegyzet* (2007)
<http://simonp.web.elte.hu/files/kozdiff.pdf>
- [18] Laczkovich Miklós, T. Sós Vera: *Valós Analízis I.*, Typotex Kiadó (2012)
- [19] Hanka László, Zalay Miklós: *Komplex függvénytan*, Műszaki Könyvkiadó, Budapest (2003)
- [20] Halász Gábor: *Bevezető komplex függvénytan - Komplex függvénytan füzetek III.*

7. Eredetiségi nyilatkozat

NYILATKOZAT

Név: *Hajóza Kristóf*
ELTE Természettudományi Kar, szak: *Matematika*
NEPTUN azonosító: *ZFJG8A*
Szakdolgozat címe: *Numerikus analízis alkalmazása az elektronikusúdon:
a Cottrell-egyenlet megoldása*

A **szakdolgozat** szerzőjeként fegyelmi felelősségem tudatában kijelentem, hogy a
dolgozatom önálló szellemi alkotásom, abban a hivatkozások és idézések standard
szabályait következetesen alkalmaztam, mások által írt részeket a megfelelő idézés
nélkül nem használtam fel.

Budapest, 2019.05.23.

Hajóza Kristóf

a hallgató aláírása