

STABIL, EXPLICIT NUMERIKUS ALGORITMUSOK DIFFÚZIÓS ÉS HŐVEZETÉSI PROBLÉMÁK MEGOLDÁSÁRA

Majár János 

egyetemi docens, Miskolci Egyetem, Fizikai és Elektrotechnikai Intézet, Fizikai Tanszék
3515 Miskolc, Miskolc-Egyetemváros, e-mail: majar.janos@gmail.com

Pszota Gábor 

egyetemi docens, Miskolci Egyetem, Fizikai és Elektrotechnikai Intézet, Fizikai Tanszék
3515 Miskolc, Miskolc-Egyetemváros, e-mail: fizpszo@uni-miskolc.hu

Absztrakt

Ebben a cikkben főként nemrég publikált explicit és stabil numerikus módszereket mutatunk be és hasonlítunk össze egymással és egy implicit módszerrel a lineáris hővezetési vagy diffúziós egyenlet megoldására. A standard explicit módszerektől (mint pl. FTCS vagy Runge-Kutta) eltérően ezek a módszerek az időlépés nem polinomiális kifejezését használják a változó új értékeinek kiszámításához. A módszerek teljesítményét olyan nem triviális analitikus megoldások reprodukálásán keresztül hasonlítjuk össze, ahol a diffúziós együttható függ a térváltozótól. A vizsgált esetekben a legjobb új módszerek sokkal jobb teljesítményt nyújtanak, mint a hagyományos Runge-Kutta módszerek.

Kulcsszavak: hővezetési egyenlet, diffúziós egyenlet, explicit módszerek, merev egyenletek; feltétel nélküli stabilitás

Abstract

In this paper, we present and compare mainly recently published explicit and stable numerical methods with each other and with an implicit method for solving the linear heat conduction or diffusion equation. Unlike standard explicit methods (such as FTCS or Runge-Kutta), these methods use a non-polynomial expression of the time step size to calculate the new values of the variable. The performance of the methods is compared via such non-trivial analytical solutions where the diffusion coefficient depends on the spatial variable. In the cases studied, the best new methods perform much better than the traditional Runge-Kutta methods.

Keywords: heat equation, diffusion equation, explicit methods, stiff equations, unconditional stability

1. Bevezetés

Közismert, hogy az egyik legegyszerűbb típusú transzport-folyamat a diffúzió, és az ezzel matematikailag analóg (Kovács and Majár, 2018) hővezetés, ami szilárd testekben zajlik. A diffúzió részecske-transzportot, a hővezetés energia-transzportot jelent. Ezeket a folyamatokat egy másodrendű lineáris parabolikus parciális differenciálegyenlet (PDE), az úgynevezett hővezetési egyenlet vagy diffúzióegyenlet írja le. Egy dimenzióban a reguláris diffúzióegyenlet a következő parciális differenciálegyenlet:

$$\frac{\partial u(x,t)}{\partial t} = D \frac{\partial^2 u(x,t)}{\partial x^2} \quad (1)$$

ahol x és t független valós változók, $u = u(x,t)$ a keresett függvény, amely fizikai jelentése koncentráció, vagy – a hővezetése esetén – hőmérséklet, D pedig a diffúziós együttható.

Ezzel az egyenlettel (és általánosításaival, mint például az advekcio-diffúzió egyenlet) szokták modellezni a diffúz részecsketranszportot számos különböző fizikai, kémiai és biológiai problémában, pl. félvezetőkben (Blaj et al., 2017) vagy épp az emberi szervezetben (Le Bihan, 2014). Emellett az egyenlet használható az épületek falain keresztül történő hővezetés szimulálására (Kareem Jalghaf et al., 2022), vagy a nedvesség szivárgásának modellezésére réteges struktúrákban (Yu et al., 2022).

Az (1) egyenlet ránézésre nagyon egyszerű, így felmerül a kérdés, hogy miért nem analitikusan oldjuk meg. Valóban, ennek az egyenletnek és még az általánosításainak is sok analitikus megoldása ismert, köztük újak is (Mojtabi and Deville, 2015; Barna et al., 2020; Barna and Kersner, 2010; Mátyás and Barna, 2021; Bastani and Salkuyeh, 2012; Agbavon et al., 2019). Sajnos azonban ezek többsége fizikailag homogén rendszerekre vonatkozik, ahol az együtthatók, pl. a D , állandók. A nagyon kevés ellenpélda egyike Zoppou és Knight eredménye (Zoppou and Knight, 1999), amely analitikus megoldásokat ad az advekcio-diffúzió egyenletre az együtthatók között fennálló speciális összefüggések esetére.

A műszaki szempontból érdekes rendszerekben a fizikai tulajdonságok, mint a fahő és a termikus diffúzióviszony, nagymértékben és bonyolult módon változhatnak kis távolságokon belül (Lienhard IV and Lienhard V, 2017; Zimmerman, 2018) (p. 15), gondoljunk csak egy mikroprocesszorra. Ezért a rendszert leíró mátrix elemeinek és így sajátértékeinek abszolút értéke számos nagyságrenden keresztül változhat, ami azt jelenti, hogy a probléma erősen merev (angolul: „stiff”) is lehet. Azonban a hagyományos explicit módszerek csak feltételesen stabilak, vagyis az időlépésnek kisebbnek kell lennie az ún. CFL küszöbszámtól. Ez a küszöb stiff rendszerekre többnyire nagyon alacsony, tehát a standard explicit módszerek az ilyen jellegű problémák megoldására nem megfelelőek, mivel túl kicsi időlépéseket kell használni. Ezért szinte kizárólag a jó stabilitási tulajdonságokkal rendelkező implicit módszereket alkalmazzák (Mascagni, 1990; Mbroh and Munyakazi, 2021). Csakhogy ezek nem egyenletről egyenletre haladnak a megoldás során, hanem egyszerre oldják meg az algebrai egyenletrendszert minden időlépés során, leggyakrabban iteratív módszerekkel. Ez kis rendszerekre gyorsan megy, ha viszont a mátrix nagy méretű, és egynél több térdimenzió van, akkor ezek az eljárások rendkívül idő- és memóriaigényesek.

Cikkünk egyik célja, hogy felhívja a figyelmet arra, hogy léteznek jobb stabilitási tulajdonságokkal rendelkező explicit módszerek. Ezek bár nem teljesen közismertek, számos kutatócsoport foglalkozik velük, lásd pl. (Appadu, 2017; Karahan, 2007; Sanjaya and Mungkasi, 2017; Pourghanbar et al., 2020; Harley, 2010; Al-Bayati et al., 2011; Nwaigwe, 2022; Savović et al., 2022; Berger et al., 2020). Vanak köztük évtizedek óta ismert ilyen módszerek, például a Dufort-Frankel vagy az odd-even hopscotch („páratlan páros ugróiskola”). Az utóbbi években azonban számos új változat is megjelent (Kovács and Gilicz, 2018; Kovács, 2020a; Saleh et al., 2020a; Saleh et al., 2020b; Nagy et al., 2021a; Nagy et al., 2021b; Kovács et al., 2021; Jalghaf et al., 2021; Kovács, 2020b; Kovács et al., 2022). Cikkünk magyar nyelven először mutat be nagyszámú ilyen módszert és azokat a teszteket, amelyek a teljesítményüket összehasonlítják. Az eredeti angol nyelvű cikkekben többnyire analitikusan bizonyították a módszerek tulajdonságait és numerikus kísérletekkel demonstrálták, hogy sokkal gyorsabban lehet velük diffúzió-jellegű problémákat megoldani, mint úgy, ha a térbeli diszkretizációt követően a

közismert MATLAB rutinokat, pl. az ode45-öt használnánk. Legutóbb megjelent egy olyan angol nyelvű cikk (Saleh et al., 2022), amelyben 14 ilyen módszert mutatnak be és tesztelnek. Ennek során olyan nem triviális analitikus megoldást használtak referenciának, amely térfüggő diffúziós együtthatóra vonatkozik. A mi cikkünk e munka folytatásának tekinthető, és egyik fő újdonsága, hogy a legelterjedtebb implicit módszerek egyikével is összehasonlítjuk az explicit módszereket. A másik, hogy az analitikus megoldások közül nem a Whittaker M, hanem a Whittaker W függvényt használjuk, amely a Kummer U speciális függvényt tartalmazza. Továbbá, teljesen más paraméterekre tesztelünk, például egy olyan esetre is, amely nagyságrendekkel merevebb minden eddigi rendszernél, amelyeken hasonló módszereket teszteltek. A tesztek során azt vizsgáljuk, melyik módszer hogyan teljesít, vagyis milyen időlépcsőkre mekkora pontossággal reprodukálja ezt a megoldást.

2. Az egyenlet és diszkretizálása inhomogén közegben

Ha a diffúzió vagy hővezetés nem homogén közegben zajlik, a következő egyenletet alkalmazhatjuk:

$$c(x)\rho(x)\frac{\partial u}{\partial t} = \frac{\partial}{\partial x}\left(k(x)\frac{\partial u}{\partial x}\right) \quad (2)$$

ahol hővezetés esetén $u = u(\vec{r}, t)$ a hőmérséklet (részecske-diffúzió esetén koncentráció), $D = k / (c\rho)$ a termikus diffuzivitás, k , c és ρ pedig rendre a hővezetési tényező, a fajhő és a (tömeg)sűrűség, melyek közül egyik sem lehet negatív.

A hely szerinti deriváltakat és a $k(x)$ függvényt egyszerre diszkretizáljuk, előbbit a leggyakoribb centrális differencia-séma felhasználásával. Ekkor

$$c(x_i)\rho(x_i)\frac{\partial u}{\partial t}\Big|_{x_i} = \frac{1}{\Delta x}\left[k\left(x_i + \frac{\Delta x}{2}\right)\frac{u(x_i + \Delta x) - u(x_i)}{\Delta x} + k\left(x_i - \frac{\Delta x}{2}\right)\frac{u(x_i - \Delta x) - u(x_i)}{\Delta x}\right].$$

Ezen a ponton csomóponti változókról átváltunk cella változókra. Ez azt jelenti, hogy u_i , c_i , és ρ_i az i cella átlaghőmérsékletének, fajhőjének és sűrűségének a közelítő értékei a cellaközépponti értékük alapján. Továbbá $k_{i,i+1}$ az i cella és a (jobb oldali) szomszédja közötti hővezetési tényező, a cellák határán lévő értékével becsülve. Az előző képlet immár az alábbi alakot ölti:

$$\frac{du_i}{dt} = \frac{1}{c_i\rho_i\Delta x}\left(k_{i,i+1}\frac{u_{i+1} - u_i}{\Delta x} + k_{i-1,i}\frac{u_{i-1} - u_i}{\Delta x}\right)$$

Egy nem egyenletesen felosztott rács esetén Δx_i jelölje a cella hosszát. A két szomszédos cella középpontja közötti távolság ekkor $d_{i,i+1} = (\Delta x_i + \Delta x_{i+1}) / 2$. Ezek felhasználásával a fenti egyenlet a következőképpen általánosítható:

$$\frac{du_i}{dt} = \frac{1}{c_i\rho_i\Delta x_i}\left(k_{i,i+1}\frac{u_{i+1} - u_i}{d_{i,i+1}} + k_{i,i-1}\frac{u_{i-1} - u_i}{d_{i,i-1}}\right)$$

Mivel egy dimenziós esetet vizsgálunk, a keresztmetszetet egységnyinek tekintjük, vagyis a cellák térfogata $V_i = \Delta x_i$. A cellák hőkapacitása $C_i = c_i\rho_i\Delta x_i$, a hővezetési ellenállások pedig a $R_{i,i+1} \approx d_{i,i+1} / k_{i,i+1}$

képlettel közelíthetőek. Ezen definíciók behelyettesítésével kapunk egy közösleges differenciálegyenlet-rendszert, amely megadja az egyes cella-hőmérsékletek időderiváltjait:

$$\frac{du_i}{dt} = \frac{u_{i-1} - u_i}{R_{i-1,i}C_i} + \frac{u_{i+1} - u_i}{R_{i+1,i}C_i} \quad (3)$$

Mátrixos alakban ez a következőképpen írható fel:

$$\frac{d\vec{u}}{dt} = M\vec{u}, \quad (4)$$

ahol az M tridiagonális mátrix főátlón kívüli elemei nem negatívak és az $m_{i,i+1} = 1/C_i R_{i,i+1}$ és $m_{i,i-1} = 1/C_i R_{i,i-1}$ képletekkel számolhatók, m_{ii} pedig ezen két elem összege negatív előjellel.

Az effajta diszkretizálásról a (Munka and Pápay, 2001) könyv 5. fejezetében és a (Nagy et al., 2021a) cikkben található további részletek.

A diffúziós együtthatóra a $D = \bar{D}x^m$ hatványfüggvény szerinti helyfüggést csak a k -k szintjén alkalmazzuk, ezért az egyszerűség kedvéért a $c \equiv 1$ és $\rho \equiv 1$ értékeket használjuk. Tekintsük az $x \in [x_0, x_{N_x}]$, $L = x_{N_x} - x_0$ intervallumot, és ezen a következő eljárással készítünk egy nem egyenletesen felosztott térbeli hálót. Kezdjük a cellahatárok x_0, x_1, \dots, x_N koordinátáinak meghatározásával:

$$x_j = x_{j-1} + \Delta x_{j-1}, \quad C_j = \Delta x_j = \Delta x_0 (1 + \gamma j), \quad j = 1, \dots, N, \quad \gamma \in \mathbb{R},$$

ahol x_0 , Δx_0 és γ a konkrét példában lesz megadva. Ha γ pozitív vagy negatív, akkor a cellák mérete értelemszerűen balról jobbra nő, illetve csökken. Ha γ értéke nulla, akkor a rács egyenközű és $L = N\Delta x$. Így az X_1, \dots, X_N cellaközéppontok az alábbiak szerint adhatók meg:

$$X_j = x_{j-1} + \frac{\Delta x_j}{2}, \quad j = 1, \dots, N$$

Az ellenállások kiszámítása a következőképpen történik:

$$R_{i,i+1} = \frac{X_{i+1} - X_i}{k_{i,i+1}} = \frac{X_{i+1} - X_i}{\bar{D}(x_i)^m}, \quad i = 1, \dots, N-1$$

Így a koncentrációkat, pl. a Dirichlet-határfeltételeket az X_1, \dots, X_N cellaközéppontokban, míg a vezetőképességeket az x_1, \dots, x_{N-1} cellahatárokon számítjuk. Az időváltozó mindig egyenletesen diszkretizált, tehát ha $t \in [t^0, t^{\text{fin}}]$, akkor

$$t^j = t^0 + jh, \quad j = 1, \dots, T, \quad hT = t^{\text{fin}} - t^0$$

3. A vizsgált módszerek

Most ismertetjük a vizsgált módszereket az (3) ODE rendszer megoldására. Először az összefüggészet a legegyszerűbb esetre (egyenletesen felosztott térbeli háló és homogén anyagtulajdonságok, vagyis az (1) egyenlet esetére) mutatjuk be, majd ezt azonnal általánosítjuk egy nem egyenletes hálóra, vagyis a (3) egyenletre. Az első, legegyszerűbb alak célja az összehasonlítás megkönnyítése, mivel a legtöbb matematikai tankönyvben és cikkben a numerikus algoritmusok ebben az alakban szerepelnek. A második, általánosabb alakok azért lényegesek, mert a jelen munkánk során csak ezeket használjuk.

Az általános egydimenziós háló esetében vezessünk be két jelölést (ezeket a cikkben gyakran fogjuk használni):

$$r_i = \frac{h}{C_i} \left(\frac{1}{R_{i,i-1}} + \frac{1}{R_{i,i+1}} \right) \text{ and } A_i = \frac{h}{C_i} \left(\frac{u_{i-1}}{R_{i,i-1}} + \frac{u_{i+1}}{R_{i,i+1}} \right), \quad i = 1, \dots, N$$

Az első mennyiség a szokásos $r = \frac{Dh}{\Delta x^2}$ hálóarány általánosítása, amely az (1) egyenletre érvényes, ha azt egyenletes hálóval diszkrétizáljuk. A második mennyiség az i cella szomszédjainak állapotát és hatását tükrözi.

1. Az első és talán legegyszerűbb algoritmus a konstans szomszéd (CNe) módszer. Ennél azt a közelítést alkalmazzuk, hogy amikor egy T_i változó új értékét kiszámítjuk, akkor elhanyagoljuk, hogy a h időlépés közben más változók is változnak, ezért kapta a „konstans szomszéd módszer” nevet (Kovács, 2020a). Ez azt jelenti, hogy u_j állandónak tekintendő ha $j \neq i$, tehát független (szétválasztott), lineáris közönséges differenciálegyenleteket kell megoldanunk. Ezt jelen esetben analitikusan is könnyen meg lehet tenni. Egyenközű rácsnál az alábbi formulát kell alkalmaznunk minden rácspontra:

$$u_i^{n+1} = u_i^n \cdot e^{-2r} + \frac{u_{i-1}^n + u_{i+1}^n}{2} (1 - e^{-2r}),$$

az általános esetben pedig a cella-változók új értékeit a következő módon számoljuk:

$$u_i^{n+1} = u_i^n \cdot e^{-r_i} + \frac{A_i}{r_i} (1 - e^{-r_i})$$

A következő lépésben ezen elsőrendű módszer által kapott értékeket felhasználva egy pontosabb módszert kapunk.

2. A CpC algoritmus (Kovács et al., 2021) a CNe séma kétfázisú módszerré szervezése, amelyben az első fázis egy ph hosszúságú töredék időlépés. Itt csak a $p = 1/2$, értéket használjuk, mert ez jellemzően nagyobb pontosságot ad, mint a p egyéb értékei. Az első fázisban az u új prediktor értékeit számítjuk ki a CNe képlettel egy $h_1 = h/2$ méretű időlépés alkalmazásával:

$$u_i^{\text{pred}} = u_i^n \cdot e^{-r} + \frac{u_{i-1}^n + u_{i+1}^n}{2} (1 - e^{-r}) \quad \text{és} \quad u_i^{\text{pred}} = u_i^n e^{-r_i/2} + \frac{A_i}{r_i} (1 - e^{-r_i/2}).$$

Ezen eredmények felhasználásával az A_i mennyiségek új értékeit számítjuk ki

$$A_i^{\text{new}} = \frac{h_1}{C_i} \left(\frac{u_{i-1}^{\text{pred}}}{R_{i,i-1}} + \frac{u_{i+1}^{\text{pred}}}{R_{i,i+1}} \right)$$

majd a második fázisban ezeket használjuk a teljes méretű időlépéshez tartozó korrekciós lépés során. Ez azt jelenti, hogy az időlépés végén a végső értékek az alábbiak:

$$u_i^{n+1} = u_i^n \cdot e^{-2r} + \frac{u_{i-1}^{\text{pred}} + u_{i+1}^{\text{pred}}}{2} (1 - e^{-2r}) \quad \text{és} \quad u_i^{n+1} = u_i^n \cdot e^{-r_i} + \frac{A_i^{\text{new}}}{r_i} (1 - e^{-r_i})$$

3. A lineáris-szomszéd módszerek (Kovács, 2020b) az 1) pontban ismertetett konstans-szomszéd módszert használják prediktornak. Ez után a korrekciós lépés során azzal a realiztikusabb feltételezéssel élnek, hogy az aktuális i -edik cella szomszédos u_j változói **lineárisan** változnak. A változás meredeksége az (1) egyenlet esetén az $s_i = u_{i-1}^{\text{pred}} + u_{i+1}^{\text{pred}} - u_{i-1}^n - u_{i+1}^n$ kifejezéssel arányos. Ennek a közelítésnek az alkalmazásával kapunk egy új, szétcsatolt ODE rendszert, amely szintén megoldható analitikusan. Az időlépés végén ennek a megoldását használjuk, hogy megadjuk számunkra a korrigált értékeket:

$$u_i^{n+1} = u_i^n e^{-2r} + \frac{u_{i-1}^n + u_{i+1}^n}{2} (1 - e^{-2r}) + \frac{s_i}{2} \left(1 - \frac{1 - e^{-2r}}{2r} \right) \quad (5)$$

Az általános esetben (vagyis tetszőleges egy dimenziós térbeli rács esetén) a prediktor lépés eredményeinek a felhasználásával új A_i^{new} értékeket kell kiszámolni:

$$A_i^{\text{new}} = \frac{h}{C_i} \left(\frac{u_{i-1}^{\text{pred}}}{R_{i,i-1}} + \frac{u_{i+1}^{\text{pred}}}{R_{i,i+1}} \right)$$

A korrektor lépés most már az alábbi alakban írható fel:

$$u_i^{n+1} = u_i^n e^{-r_i} + \left(A_i - \frac{A_i^{\text{new}} - A_i}{r_i} \right) \frac{1 - e^{-r_i}}{r_i} + \frac{A_i^{\text{new}} - A_i}{r_i} \quad (6)$$

4. Az (5) vagy (6) egyenlet korrektor értékeit használva megismételhető (5) vagy (6) - nem egyenletes rács esetén először A_i^{new} újraszámításával - az új korrektor eredmények megkapásához. Ez az eljárás egy összességében három fázisból álló sémát ad, amelyet LNe3 módszernek nevezünk (Kovács, 2020b). Ez az algoritmus még mindig másodrendű, de pontosabb, mint az LNe2.

5. A CLL módszer (Kovács et al., 2022) nagyon hasonló az LNe3 módszerhez. A különbség az, hogy az első és a második fázisban lévő töredék időlépések miatt harmadrendű időbeli konvergenciát ér el, de csak akkor, ha a második tört időlépés $h_2 = 2h / 3$ hosszú. Általában az első tört lépés hossza ph , $2/3 \leq p < 2$, de itt a legjobb pontosság elérése és a CPU-idő megtakarítása érdekében a $p = 2/3$ értéket választjuk, elkerülve azt is, hogy az e^{-pr_i} exponenciális tényezőket még egyszer ki kelljen számítani. Tehát az első fázisban a változók új prediktor értékeit számítjuk ki a CNe képletrel, de egy $h_1 = 2h / 3$ időlépéssel:

$$u_i^C = u_i^n \cdot e^{-4r/3} + \frac{u_{i-1}^n + u_{i+1}^n}{2} (1 - e^{-4r/3}) \quad \text{és} \quad u_i^C = u_i^n e^{-2r_i/3} + \frac{A_i}{r_i} (1 - e^{-2r_i/3}).$$

A második fázisban az első korrektor értékek megkapásához a (5) és (6) képletekhez hasonló összefüggéseket használunk, de a $h_2 = 2h/3$ időlépéssel. Az egyenletes háló esetére a következőt kapjuk:

$$u_i^{n+1} = u_i^n e^{-4r/3} + \frac{u_{i-1}^n + u_{i+1}^n}{2} (1 - e^{-4r/3}) + \frac{s_i^1}{2} \left(1 - \frac{1 - e^{-4r/3}}{4r/3} \right)$$

ahol $s_i^1 = u_{i-1}^C + u_{i+1}^C - u_{i-1}^n - u_{i+1}^n$. Nem egyenletes háló esetén új A_i^{new} értékeket kell számolnunk az

$$A_i^C = \frac{h_1}{C_i} \left(\frac{u_{i-1}^C}{R_{i,i-1}} + \frac{u_{i+1}^C}{R_{i,i+1}} \right)$$

képlettel. Ezeket használva a korrektor lépés a következő alakot ölti:

$$u_i^{n+1} = u_i^n e^{-2r_i/3} + \left(A_i - \frac{A_i^C - A_i}{2r_i/3} \right) \frac{1 - e^{-2r_i/3}}{r_i} + \frac{A_i^C - A_i}{r_i}$$

A harmadik fázisban egy teljes időlépést veszünk. Az egyenletes háló esetére a következőt kapjuk:

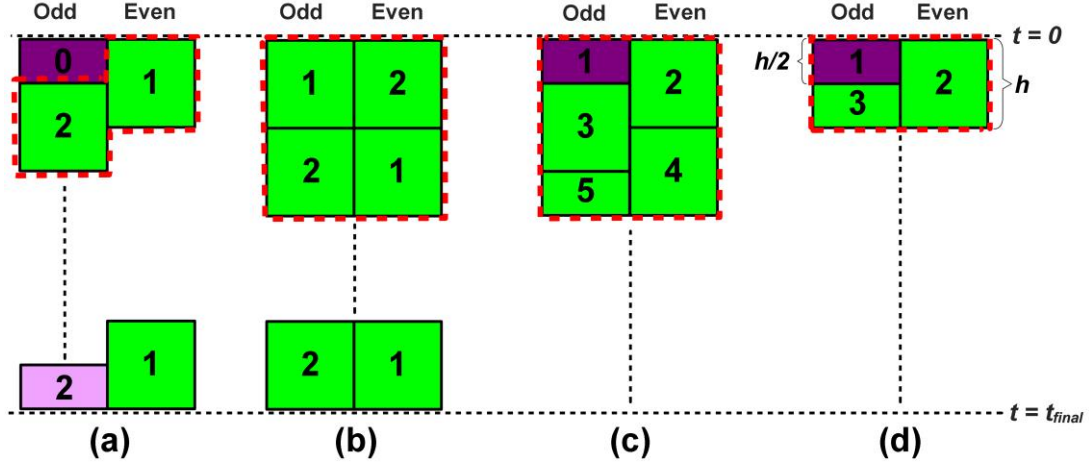
$$u_i^{n+1} = u_i^n e^{-2r} + \frac{u_{i-1}^n + u_{i+1}^n}{2} (1 - e^{-2r}) + \frac{3s_i^2}{4} \left(1 - \frac{1 - e^{-2r}}{2r} \right)$$

ahol $s_i^2 = u_{i-1}^{\text{CL}} + u_{i+1}^{\text{CL}} - u_{i-1}^n - u_{i+1}^n$. Az általánosabb esetben pedig az alábbi eredményt kapjuk:

$$u_i^{n+1} = u_i^n e^{-r_i} + \left(A_i - \frac{A_i^{\text{CL}} - A_i}{2r_i/3} \right) \frac{1 - e^{-r_i}}{r_i} + \frac{A_i^{\text{CL}} - A_i}{2r_i/3}$$

ahol $A_i^{\text{CL}} = \frac{h_2}{C_i} \left(\frac{u_{i-1}^{\text{CL}}}{R_{i,i-1}} + \frac{u_{i+1}^{\text{CL}}}{R_{i,i+1}} \right)$.

Most rátérünk a hopscotch módszerek bemutatására. A páratlan-páros hopscotch módszer bármelyik verziójának alkalmazásához kétrétű (sakktábla-szerű) rácsra van szükség, amelyben a páratlan csomópontok vagy cellák minden legközelebbi szomszédja páros és fordítva. A vizsgált sémák térbeli és időbeli szerkezetét az 1. ábra mutatja be. Az ismétlődő blokkokat szaggatott piros vonal jelzi. Például a leapfrog-hopscotch struktúra két fél és sok teljes időlépésből áll. Először egy feles méretű időlépést (amelyet a 1a. ábrán egy lila téglalap jelképez, benne a '0' számmal) veszünk a páratlan cellák esetében a kezdeti értékeket használva. Ez után teljes időlépéseket haladunk (zöld négyzetek) szigorúan felváltva a páros és a páratlan cellák esetében az utolsó időlépés végéig (rózsaszín téglalap), amelyet meg kell felezni a páratlan celláknál, hogy azok pontosan ugyanazt a végső időt ériék el, mint a páros csomópontok. A lényeg az, hogy az u_i új értékének kiszámításakor mindig az $u_{i\pm 1}$ szomszédok legfrissebb értékeit kell használni, ami biztosítja a stabilitást és egyben elég gyors konvergenciát.



1. ábra. Az hopscotch típusú módszerek tér-idő szerkezete a páratlan (odd) és a páros (even) rácspontokra. a) Leapfrog-hopscotch (LH), (b) eredeti páratlan-páros hopscotch (OOEH), (c) eltolt hopscotch (SH), (d) aszimmetrikus hopscotch (ASH).

6. Az LH-CNe címkével jelölt leapfrog (bakugrás)-hopscotch-CNe módszert (Nagy et al., 2021b) úgy kapjuk meg, hogy a CNe képletet alkalmazzuk az LH struktúra minden fázisában a megfelelő nagyságú időlépéssel. Például az első fázis (amely fél időlépcső hosszú az időtengelyen) az alábbi módon néz ki:

$$u_i^{n+1} = u_i^n \cdot e^{-r} + \frac{u_{i-1}^{\text{latest}} + u_{i+1}^{\text{latest}}}{2} (1 - e^{-r}) \quad \text{és} \quad u_i^{n+1} = u_i^n \cdot e^{-r_i/2} + \frac{A_i^{\text{latest}}}{r_i} (1 - e^{-r_i/2})$$

egy egyenletes és egy nem egyenletes háló esetén. Mindig a szomszédok legfrissebb értékeit használjuk az A_i^{latest} kiszámításához, hasonló képlettel, mint korábban az A_i^{new} értékeket. A programozás egyszerűsége érdekében az 1D rács összes cellájának N száma mindig páratlan, így az első fázisban az $i = 3, 5, 7, N - 2$ cellákra a fenti képletet alkalmaztuk, majd a peremfeltételeket kiszámítottuk az időintervallumok közepére az első és az utolsó cellára egyaránt.

7. A következő módszer a leapfrog-hopscotch (LH) módszer. Ez az úgynevezett θ képletet használja. Esetünkben ez az alábbi módon néz ki egyenletes hálóra:

$$u_i^{n+1} = \frac{(1 - 2r\theta)u_i^n + r(u_{i-1}^{\text{latest}} + u_{i+1}^{\text{latest}})}{1 + 2r(1 - \theta)} \quad (7)$$

és (az általános esetben),

$$u_i^n = \frac{(1 - r_i\theta)u_i^n + A_i^{\text{latest}}}{1 + r_i(1 - \theta)} \quad (8)$$

Az erről szóló cikkben (Nagy et al., 2021b) számos numerikus kísérlet került végrehajtásra, és azóta a legpontosabbnak bizonyult képletet használjuk (L2 a (Nagy et al., 2021b) cikkben). Ez azt jelenti, hogy $\theta = 0$ kerül alkalmazásra az első fázisban, az összes többi fázisban pedig $\theta = 1/2$.

8. Az eredeti páratlan-páros hopscotch algoritmus (Gourlay and McGuire, 1971) esetében, amelyet itt OOEH néven rövidítünk, az első fázisban a legelterjedtebb FTCS (explicit Euler) képletet, a második fázisban pedig az implicit Euler képletet használták a 3 (b) ábrán látható szerkezetben. Az egyenletes és a nem egyenletes hálók esetében a képletek az alábbiak:

$$\text{FTCS: } u_i^{n+1} = (1-2r)u_i^n + r(u_{i-1}^n + u_{i+1}^n) \quad \text{és} \quad u_i^{n+1} = (1-r_i)u_i^n + A_i.$$

$$\text{Implicit Euler: } u_i^{n+1} = \frac{u_i^n + r(u_{i-1}^{n+1} + u_{i+1}^{n+1})}{1+2r} \quad \text{és} \quad u_i^{n+1} = \frac{u_i^n + A_i^{\text{latest}}}{1+r_i}.$$

Amint az korábban bizonyítást nyert (Saleh et al., 2020a), ez egy nagyon hatékony explicit módszer térben egyenletes esetekre. Ha viszont a merevségi hányados nagy, akkor a hibája nagyon nagy is lehet (Saleh et al., 2020b).

9. A fordított (páratlan-páros) hopscotch módszer (reversed hopscotch, RH) ugyanazt a szerkezetet és képleteket alkalmazza, mint az OOEH módszer, de a képletek ellentétes sorrendben vannak. Azonban amikor az első fázis számításai elkezdődnek, a szomszédok új értékei nem ismertek. Ezért az implicit képlet csak egy trükkkel alkalmazható, mely szerint a szomszédokat nem implicit módon, hanem explicit módon kell kezelni. Ez az elgondolás az alábbi képleteket eredményezi az első fázisban:

$$u_i^{n+1} = \frac{u_i^n + r(u_{i-1}^n + u_{i+1}^n)}{1+2r} \quad \text{és} \quad u_i^{n+1} = \frac{u_i^n + A_i}{1+2r_i}$$

Ha megvan az eredeti OOEH programkódja, akkor könnyen megkapható az RH algoritmus kódja, hiszen csak az első és a második fázis képleteit kell megcserélni. Rendkívül merev rendszerek esetén ennek az RH algoritmusnak a hibái sokkal kisebbek, mint az OOEH módszeré (Saleh et al., 2020b).

10. Az eltolt hopscotch (SH) algoritmus (Nagy et al., 2021a) öt fázisból áll (két feles és három teljes időlépés). Amint az a 3 (c) ábrán látható, ezek összesen két teljes időlépést tesznek ki páratlan és páros cellák esetében egyaránt. Ebben a cikkben a (17) és (18) theta képleteket használjuk a legjobbnak bizonyult theta értékekkel (S4 algoritmus a (Nagy et al., 2021a) cikkben). Ez azt jelenti, hogy az első fázisban $\theta=0$, a második, a harmadik és a negyedik fázisban $\theta=1/2$, míg az ötödik fázisban $\theta=1$ értéket vettünk.

11. Az aszimmetrikus hopscotch (ASH) algoritmus (Saleh and Kovács, 2021) az SH séma redukált változata. Amint az a 3(d) ábrán látható, ez csak három fázisból áll (két fél és egy teljes időlépés), amelyek összesen egy teljes időlépést tesznek ki a páratlan és páros cellák esetében egyaránt. A legjobbnak bizonyult theta értékek halmaza (A1 algoritmus a (Saleh and Kovács, 2021) cikkben) $\theta=0$ az első, $\theta=1/2$ a második és $\theta=1$ a harmadik fázisban.

12. A pszeudo-implicit (PI) kétfázisú algoritmus a (Jalghaf et al., 2021) cikkből származik (ott az 5. algoritmus) $\lambda=1$ paraméterrel rendelkező tiszta diffúziós egyenlet esetén, ami azt jelenti, hogy fél időlépést veszünk a prediktor értékek megkapásához, majd egy teljes időlépést a korrektor értékekhez. A következő képleteket kell alkalmazni minden cellára:

$$1. \text{ fázis: } u_i^{\text{pred}} = \frac{u_i^n + r/2(u_{i-1}^n + u_{i+1}^n)}{1+r} \quad \text{és} \quad u_i^{\text{pred}} = \frac{u_i^n + A_i/2}{1+r_i/2}.$$

$$2. \text{ fázis: } u_i^{n+1} = \frac{(1-r)u_i^n + r(u_{i-1}^{\text{pred}} + u_{i+1}^{\text{pred}})}{1+r} \quad \text{és} \quad u_i^{n+1} = \frac{(1-r_i/2)u_i^n + A_i^{\text{new}}}{1+r_i/2}.$$

ahol $A_i^{\text{new}} = \frac{h}{C_i} \left(\frac{u_{i-1}^{\text{pred}}}{R_{i,i-1}} + \frac{u_{i+1}^{\text{pred}}}{R_{i,i+1}} \right)$. Látható, hogy a szomszédok explicit kezelésének trükkje ugyanaz,

mint az RH módszernél.

13. A Dufort–Frankel (DF) módszer (Hirsch, 1988) (313. o.) egy ismert, de nem szokványos explicit és feltétel nélkül stabil algoritmus, amelynek a képlete az egyenletes és nem egyenletes esetre:

$$u_i^{n+1} = \frac{(1-2r)u_i^{n-1} + 2r(u_{i-1}^n + u_{i+1}^n)}{1+2r} \quad \text{és} \quad u_i^{n+1} = \frac{(1-r_i)u_i^{n-1} + 2A_i}{1+r_i}.$$

Mivel a képletek tartalmazzák az u_i^{n-1} tagot, ez egyfázisú, de kétlépéses módszer. Ez azt jelenti, hogy az algoritmus elindításához az u_i^1 egy másik módszerrel számítandó ki az u_i^0 értékéből. Erre a célra a CNe képletet használjuk.

14. Összehasonlítás céljából a negyedrendű Runge-Kutta (RK4) módszernek azt a változatát használjuk, amely talán a legelterjedtebb, lásd pl. (Chapra and Canale, 2015), 737. o. Ha ezt a térben diszkrétizált rendszerünkre alkalmazzuk, akkor a következőt kapjuk:

$$\begin{aligned} k_i^1 &= r(u_{i-1}^n + u_{i+1}^n - 2u_i^n) & \text{és} & \quad k_i^1 = A_i - r_i u_i^n, \\ k_i^2 &= r(u_{i-1}^n + k_{i-1}^1/2 + u_{i+1}^n + k_{i+1}^1/2 - 2u_i^n - k_i^1) & \text{és} & \quad k_i^2 = A_i^1 - r_i(u_i^n + k_i^1/2), \\ k_i^3 &= r(u_{i-1}^n + k_{i-1}^2/2 + u_{i+1}^n + k_{i+1}^2/2 - 2u_i^n - k_i^2) & \text{és} & \quad k_i^3 = A_i^2 - r_i(u_i^n + k_i^2/2), \\ k_i^4 &= r(u_{i-1}^n + k_{i-1}^3 + u_{i+1}^n + k_{i+1}^3 - 2u_i^n - 2k_i^3) & \text{és} & \quad k_i^4 = A_i^3 - r_i(u_i^n + k_i^3/2), \end{aligned}$$

és végül

$$u_i^{n+1} = u_i^n + (k_i^1 + 2k_i^2 + 2k_i^3 + k_i^4)/6.$$

$$\text{Itt } A_i^s = \frac{h}{C_i} \left(\frac{u_{i-1}^n + k_{i-1}^s/2}{R_{i,i-1}} + \frac{u_{i+1}^n + k_{i+1}^s/2}{R_{i,i+1}} \right), \quad s \in \{1, 2, 3\}.$$

Látható, hogy az eddig ismertetett sémák explicitek, az új értékek kiszámíthatók egy egyenletrendszer megoldása nélkül. A bemutatott sémák, a DF kivételével egylépéses módszerek abban az értelemben, hogy az új értékek kiszámítása után a változók korábbi értékeit nem használjuk és így nem is szükséges tárolni őket.

Ezen a ponton érdemes egy megjegyzést tennünk. Az explicit RK módszerek, köztük az RK4 is a h időlépcső polinomiális kifejezését használják az u változó új értékeinek kiszámításához, míg az 1-13 pontban ismertetettek nem. Előbbi esetben mindenképp jelen lesz az instabilitás veszélye, mivel a változó elég nagy értékeire minden polinom divergál.

15. Az implicit Euler séma az eredeti (Saleh et al., 2022) cikkben sajnos nem szerepelt. Ebben a cikkben azért vettük bele az összehasonlításba, mert rendkívül gyakran használják és ezért alapvető fontosságú, hogy legyen információnk arról, milyen teljesítményt nyújt a többi vizsgált módszerhez képest. Az implicit sémát úgy implementáltuk, hogy invertáljuk a (4) egyenletben a mátrixot és ezzel szorozzuk lépésről lépésre az u vektort.

Tehát a CNe, CpC, LNe, LNe3, CLL, LH-CNe, LH, RH, SH, ASH és PI módszereket kutatócsoportunk hozta létre, és a verifikációk, az időnként hosszadalmas analitikus bizonyítások stb. jellemzően ezekben az eredeti cikkekben kerültek bemutatásra. A CNe módszer elsőrendű, a CpC, LNe, LNe3, LH-CNe, LH, OOEH, RH, SH, ASH, PI és a DF módszerek másodrendűek, a CLL módszer harmadrendű, míg az RK4 séma negyedrendű az időlépés méretét tekintve. Minden itt vizsgált algoritmus (természetesen az RK4 módszer kivételével) feltétel nélkül stabil a lineáris diffúziós egyenletre, vagyis a bevezetőben említett CFL határérték nem érvényes rájuk. Ismételten hangsúlyozzuk, hogy a feltétel nélküli stabilitás nem szabály, hanem kivétel az explicit módszerek között. Köztudott, hogy például az explicit Runge-Kutta módszerek nem lehetnek A-stabilak (Iserles, 2009) (60. o.). Érdemes megjegyezni, hogy a CNe, CpC, LNe, LNe3 és LH-CNe sémák nemcsak stabilak, hanem a pozitivitást is megőrzik, ami határt szab a hibájuknak. Ez azt is jelenti, hogy nem eredményeznek fizikailag értelmezhetetlen oszcillációkat még nagyon nagy időlépések esetén sem. Azonban, mint látni fogjuk, ez korlátozza konvergenciájuk gyorsaságát, így gyakran ezek a legkevésbé pontosak kis és közepes méretű időlépések esetén. A hopscotch típusú módszerek speciális sakktáblaszerű rácsot igényelnek, de egy dimenzióban ez egy triviális követelmény. Ezeknek az algoritmusoknak azonban nincs szükségük az u változóra vonatkozó tömb egy másik példányának ideiglenes tárolására, így a memóriagigéjük minimális. Egyéb módszerek, még a CNe is, legalább egy extra tömböt tárolnak ugyanannyi elemmel, mint az u .

4. Összehasonlítás analitikus megoldásokkal

A verifikáció érdekében a módszereink által kapott eredményeket összehasonlítottuk pontos eredményekkel. A már korábban említett cikkben a diffúziós együttható egyszerű $D(x) = \bar{D}x^m$ esetre adtak egy analitikus megoldás-sereget. Ekkor az egyenlet a

$$\frac{\partial u(x,t)}{\partial t} = \bar{D} \frac{\partial}{\partial x} \left(x^m \frac{\partial u(x,t)}{\partial x} \right)$$

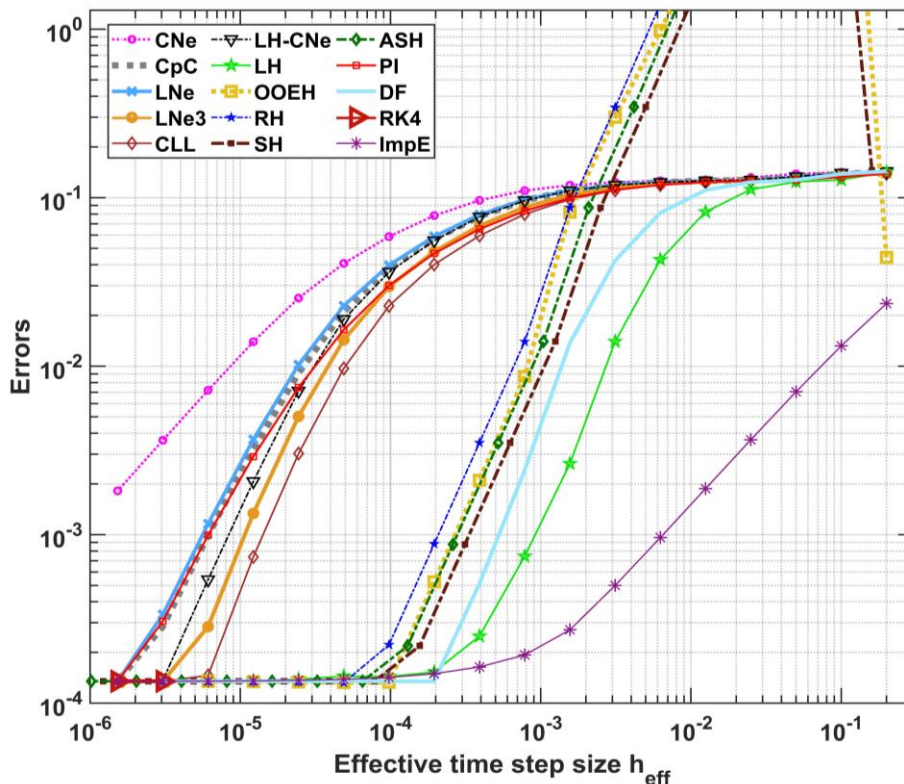
alakot ölti, ahol a \bar{D} konstans dimenziója minden m értékre éppen olyan, hogy a megfelelő eredő dimenzió kijöjjön. Az analitikus megoldás:

$$u(x,t) = \frac{t^{-\alpha}}{\sqrt{\eta}} e^{-\frac{\eta^{-s}}{2\bar{D}s^2}} \left\{ c_1 \cdot M_{\frac{(1+2s\alpha)|s|}{2s^2}, \frac{s+1}{2s}} \left(\frac{|s|\eta^{-s}}{s^3\bar{D}} \right) + c_2 \cdot W_{\frac{(1+2s\alpha)|s|}{2s^2}, \frac{s+1}{2s}} \left(\frac{|s|\eta^{-s}}{s^3\bar{D}} \right) \right\},$$

ahol $s=m-2$, $\eta = x \cdot t^{1/s} \in \square$. A megoldásban α , c_1 és c_2 tetszőleges konstansok, M és W pedig a Whittaker függvények (Olver et al., 2011; Whittaker function - wikipedia.org). Mivel az eredeti (Saleh et al., 2022) cikkben az M függvényt tartalmazó megoldást reprodukálták (tehát a $c_1 = 1$ és $c_2 = 0$ választással éltek), mi ellenkezőképp járunk el, azaz ebben a cikkben $c_1 = 0$ és $c_2 = 1$. Emellett, mint azt látni fogjuk, a paramétereket is teljesen másképp választjuk meg, mint ott, amiből újabb tanulságokat kapunk majd. A feladat az lesz, hogy reprodukáljuk ezt az analitikus megoldást t_0 és t_{FIN} között. A hiba, amit kiszámolunk, a maximális eltérés (a különbség abszolút értéke) a referencia megoldástól.

1. Teszt. Először egy közepesen merev esetet vizsgálunk meg a következő paraméter választásokkal: $m = 3.2$, $\bar{D} = 4$, $x_0 = 0.1$, $\Delta x_1 = 0.005$, $N = 299$, $\gamma = 0.005$, továbbá $t_0 = 2$, $T = 0.2$, $t_{FIN} = t_0 + T$. Így az utolsó cella $\Delta x_{299} = 0.0125$ hosszú, az ellenállások pedig 6.8-tól $5.2 \cdot 10^{-4}$ -ig csökkennek. A merevségi arány $1.7 \cdot 10^6$, az FTCS módszer küszöb-lépéshossza pedig $3.4 \cdot 10^{-6}$.

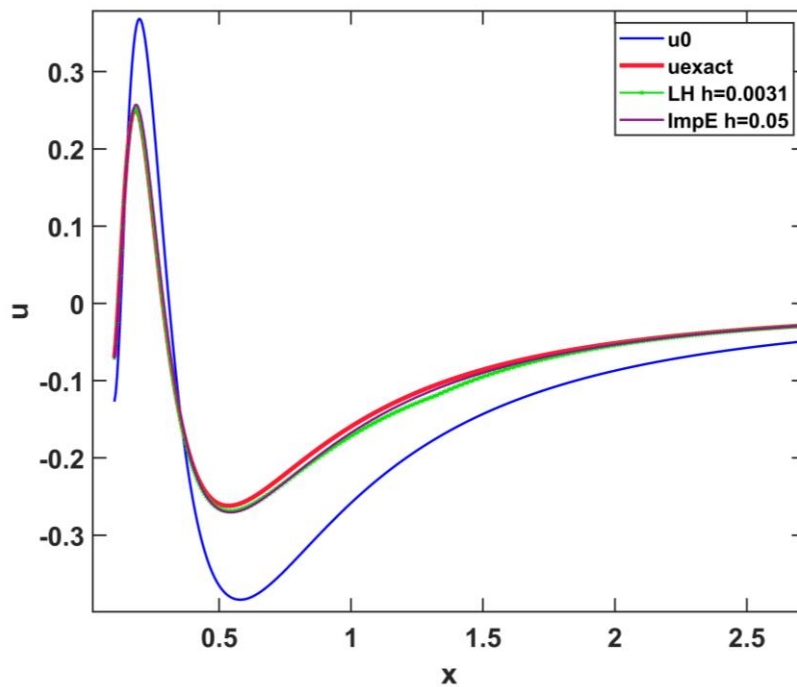
A hiba, mint az időlépés hosszának a függvénye, a 2. ábrán látható. Ebből azt olvashatjuk le, hogy bár csak elsőrendű, mégis az implicit Euler módszer konvergál a leggyorsabban. Őt követik a hopscotch típusú módszerek és a DF algoritmus. Ezek hibája már egészen megközelíti a minimális reziduális hibát (ami a tér-diszkretizációból adódik), mire a negyedrendű RK egyáltalán eredményt ad. Igaz, hogy ez utóbbi eredmény rögtön a lehető legpontosabb, de ez a pontosság a műszaki feladatoknál gyakran felesleges. Például azért, mert az input adatok pontossága sokkal kisebb, emellett többnyire semmilyen következménye sincs annak, ha az eredmény a sokadik tizedes-jegyben pontatlan.



2. ábra. A maximum hibák az effektív időlépés függvényében a vizsgált 15 módszerre az 1. tesztnél.

Mint ahogy azt a bevezetőben említettük, nagy rendszerekre az implicit módszerek igen lassúak lehetnek, de kis cellaszám esetén őket érdemes használni ilyen típusú problémák megoldására. Nagy cellaszám esetén viszont a stabil explicit módszerek valószínűleg gyorsabbak náluk, bármiféle komolyabb optimalizálás, adaptív lépésméret vezérlés vagy párhuzamosítás nélkül is. Az is látható, hogy a leapfrog-hopscotch módszeren kívül a többi hopscotch algoritmus nagyon nagy hibát ad nagy h -ra. A görbék alakjából viszont látszik, hogy ez nem valódi instabilitást jelent, mivel annál a hiba nem folyamatosan csökkenne h -val, hanem ugrásszerűen. A valódi instabilitásnál a hiba korlátlanul növekszik, ahogy az idő halad. Itt erről nincs szó, pl. a vizsgált esetben az összes hiba maximuma 634.4, az RH módszerre. Megvizsgáltuk rögzített időlépcsőre, nagyobb T értékekre ezt a hibát, és némi növekedés után (pl. $T=0.8$ -ra 797.4 volt) csökkenni kezdett ($T=12.8$ -ra pl. már csak 600.8 volt), ami ellentétes az instabilitás definíciójával. Mindez azért lehetséges, mert a vizsgált sémák a diffúzióegyenletet oldják meg, és a kialakult instabilitások, oszcillációkra is hat a diffúzió.

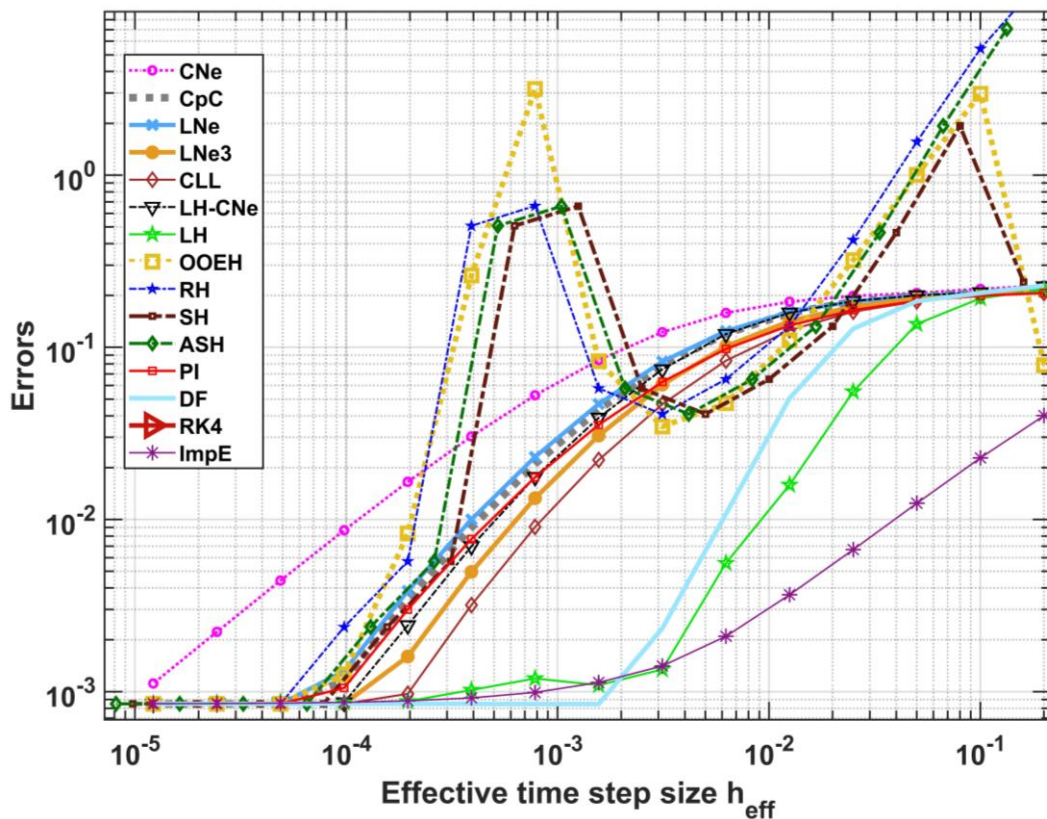
A 3. ábrán az $u(x)$ függvényt ábrázoltuk a kezdeti és a végső időpillanatban. Utóbbi esetre az analitikus megoldás mellett két numerikus megoldást is ábrázoltunk (ezek kezdeti feltételét természetesen az analitikus megoldásból számoltuk a kezdeti időpontban), ami alapján azt lehet mondani, hogy a numerikus megoldások simák, nem fizikai oszcillációk nem jelennek meg bennük.



3. ábra. A kezdeti $u(t=0, x)$ függvény, valamint az analitikus megoldás és két numerikus megoldás a szimuláció végén az 1. tesztnél.

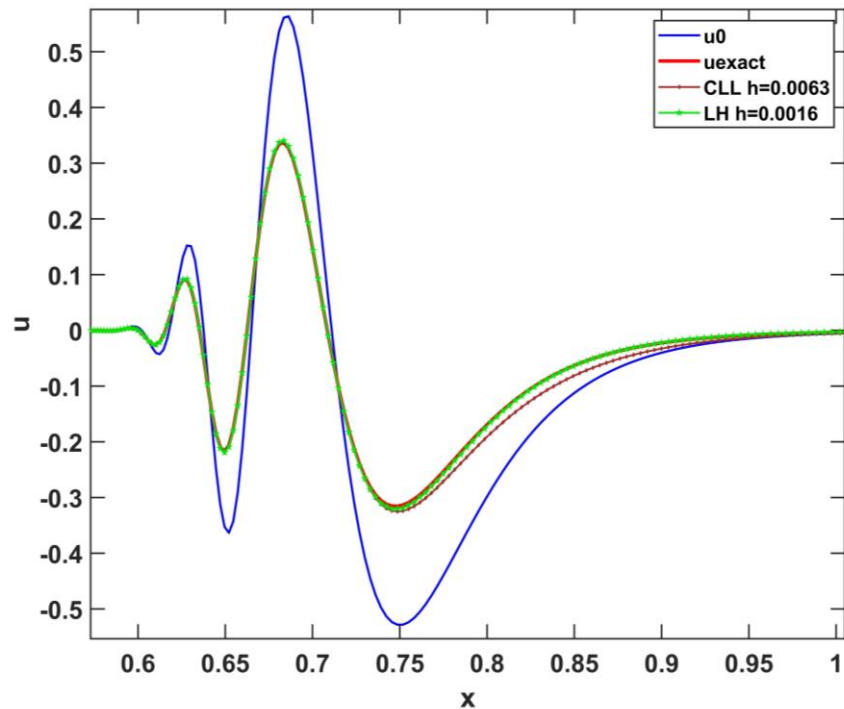
2. Teszt. Most az alábbi paramétereket állítjuk be: $m = 20$, $\bar{D} = 8$, $t_0 = 3$, $t_{\text{FIN}} = 3.2$, $x_0 = 0.5$, $\Delta x_1 = 0.0004$, $N = 499$, $\gamma = 0.05$. Így az utolsó cella $\Delta x_{499} = 0.0104$ hosszú, az ellenállások pedig 443.4-től $9.0 \cdot 10^{-13}$ -ig csökkennek. Már ebből is látható, hogy most egy rendkívül merev rend-

szert sikerült konstruálnunk, a merevségi arány $1.4 \cdot 10^{17}$. Az FTCS módszer küszöb-lépéshossza $5.5 \cdot 10^{-15}$, ami olyan kicsi, hogy a hagyományos explicit módszerek két okból is használhatatlanok. Az egyik ok, hogy párhuzamosítás nélkül ezen módszerek nagyon nagy futási idővel működnek. A másik ok, hogy a kerekítési hibák felhalmozódása az időlépések számával arányosan nő és így a kapott megoldás használhatatlanul pontatlan lenne. Hangsúlyozzuk, hogy ilyen szélsőséges esetet a fentebb hivatkozott publikációk egyikében sem vizsgáltak. A 4. ábrán a hiba-függvényekből látható, hogy a feltétel nélkül stabil explicit módszerek relatíve jól teljesítenek ebben az esetben is, hasonlóan az implicit módszerhez. Az RK4 viszont egyáltalán nem szerepel az ábrán, hiszen még 10 nagyságrenddel (!) kellett volna csökkentenünk az időlépés hosszát, hogy kapjunk egy pontot az ábrán, azaz egy használható megoldást.



4. ábra. A maximum hibák az effektív időlépés függvényében a vizsgált 15 módszerre a 2. teszténél.

Az 5. ábrán ismét a koncentráció helyfüggését mutatjuk be, de csak a vizsgált térbeli tartomány bal oldalán, azaz kis x értékekre. Az x nagyobb értékeire a függvények nagyon rásimulnak az x tengelyre. Ennek a tartománynak a vizsgálata fizikai szempontból triviális, hiszen a koncentráció lényegében nem változik, vagyis semmi sem történik a tér ezen szegmensében. Másfelől viszont itt nagyon kicsik az R ellenállások, és épp ez okozza a nagy merevségi együtthatót és a RK módszerek instabilitását. Ez azt jelenti, hogy rátapintottunk a hagyományos explicit módszerek gyenge pontjára.



5. ábra. A kezdeti $u(t=0, x)$ függvény, valamint az analitikus megoldás és két numerikus megoldás a szimuláció végén a 2. tesztél. Nagyobb x értékek esetén mindegyik függvény 0-hoz tart.

5. Összefoglalás és következtetések

Tizennégy explicit és egy implicit numerikus algoritmust teszteltünk a nem stacionárius hővezetési vagy diffúziós egyenlet megoldására. A módszerek közül 13 explicit stabil módszer, amelyből 11-et nemrég publikáltak. Tesztelésre egy olyan rendszert vizsgáltunk, ahol a diffúziós együttható függ a helytől. Ennek a problémának az analitikus megoldása a Whittaker függvényeket tartalmazza, így kifejezetten bonyolult.

Eredményeinkből az látszik, hogy minél szélsőségesebb értékeket vesz fel a diffúziós együttható, ill. a hővezetési ellenállás, annál alacsonyabb lesz a CFL küszöb. Ezt azt eredményezi, hogy az explicit Runge-Kutta módszerek egyre kevésbé használhatóak, míg a feltétel nélkül stabil módszereknél csak kis mértékben csökken a pontosság. Az implicit módszer nem érzékeny erre a változásra és kis rendszerméret, azaz alacsony cellaszám esetén ezt érdemes használni. Amennyiben viszont a cellák száma nagy, az implicit módszer aránytalanul lelassul. Itt lehet fontos szerepük az explicit stabil módszereknek, köztük elsősorban a leapfrog-hopscotch, a Dufort-Frankel és a CLL eljárásoknak. A cikkben demonstráltuk, hogy olyan merev esetekben is jól használhatók ezek a módszerek, ahol a kutatók többsége kerüli az explicit módszerek alkalmazását.

Irodalom

- [1] Kovács, E. and Majár, J.: *On some analogous transient phenomena*, 2018 MultiScience - XXXII. microCAD International Multidisciplinary Scientific Conference, Miskolc, p. C1/1.

- [2] Blaj, G., Kenney, C. J., Segal, J. and Haller, G. (2017). Analytical solutions of transient drift-diffusion in P-N junction pixel sensors. 1–13. <https://doi.org/10.2172/1443745>
- [3] Le Bihan, D. (2014). Diffusion MRI: What water tells us about the brain. *EMBO Mol. Med.*, 6(5), 569–573. <https://doi.org/10.1002/emmm.201404055>
- [4] Kareem Jalghaf, H., Omle, I., and Kovács, E. (2022). A comparative study of explicit and stable time integration schemes for heat conduction in an insulated wall. *Buildings*, 12(6), 824. <https://doi.org/10.3390/buildings12060824>
- [5] Yu, H. et al. (2022). The moisture diffusion equation for moisture absorption of multiphase symmetrical sandwich structures. *Mathematics*, 10(15), 2669. <https://doi.org/10.3390/math10152669>
- [6] Mojtabi, A., and Deville, M. O. (2015). One-dimensional linear advection-diffusion equation: Analytical and finite element solutions. *Comput. Fluids*, 107, 189–195. <https://doi.org/10.1016/j.compfluid.2014.11.006>
- [7] Barna, I. F., Bognár, G., Guedda, M., Mátyás, L., and Hriczó, K. (2020). Analytic self-similar solutions of the Kardar-Parisi-Zhang interface growing equation with various noise terms. *Math. Model. Anal.*, 25(2), 241–256. <https://doi.org/10.3846/mma.2020.10459>
- [8] Barna, I. F., and R. Kersner, R. (2010). Heat conduction: A telegraph-type model with self-similar behavior of solutions. *J. Phys. A Math. Theor.*, 43(37), 375210. <https://doi.org/10.1088/1751-8113/43/37/375210>
- [9] Mátyás, L., and Barna, I. F. (2021). General self-similar solutions of diffusion equation and related constructions. *arXiv2104.09128v1 [physics.class-ph]*
- [10] Bastani, M., and Salkuyeh, D. K. (2012). A highly accurate method to solve Fisher’s equation. *Pramana - J. Phys.*, 78, 335–346. <https://doi.org/10.1007/s12043-011-0243-8>
- [11] Agbavon, K. M., Appadu, A. R., and Khumalo, M. (2019). On the numerical solution of Fisher’s equation with coefficient of diffusion term much smaller than coefficient of reaction term. *Adv. Differ. Equations*, 146, 1–33. <https://doi.org/10.1186/s13662-019-2080-x>
- [12] Zoppou, C., and Knight, J. H. (1999). Analytical solution of a spatially variable coefficient advection-diffusion equation in up to three dimensions. *Appl. Math. Model.*, 23(9), 667–685. [https://doi.org/10.1016/S0307-904X\(99\)00005-0](https://doi.org/10.1016/S0307-904X(99)00005-0)
- [13] Lienhard IV, J. H., and Lienhard V, J. H. (2017). *A heat transfer textbook, fourth edition*. Cambridge, Massachusetts, USA: Phlogiston Press.
- [14] Zimmerman, R. W. (2018). *The imperial college lectures in petroleum engineering*. Singapore, London: World Scientific Publishing. <https://doi.org/10.1142/q0146>
- [15] Mascagni, M. (1990). Backward Euler method for numerical solution of the Hodgkin-Huxley equations of nerve conduction. *SIAM J. Numer. Anal.*, 27(4), 941–962. <https://doi.org/10.1137/0727054>
- [16] Mbroh, N. A., and Munyakazi, J. B. (2021). A robust numerical scheme for singularly perturbed parabolic reaction-diffusion problems via the method of lines. *Int. J. Comput. Math.*, 99(6), 11391158. <https://doi.org/10.1080/00207160.2021.1954621>
- [17] Appadu, A. R. (2017). Performance of UPFD scheme under some different regimes of advection, diffusion and reaction. *International Journal of Numerical Methods for Heat and Fluid Flow*, 27(7), 1412–1429. <https://doi.org/10.1108/HFF-01-2016-0038>
- [18] Karahan, H. (2007). Unconditional stable explicit finite difference technique for the advection-diffusion equation using spreadsheets. *Adv. Eng. Softw.*, 38(2), 80–86. <https://doi.org/10.1016/j.advengsoft.2006.08.001>

- [19] Sanjaya, F., and Mungkasi, S. (2017). A simple but accurate explicit finite difference method for the advection-diffusion equation. *J. Phys. Conf. Ser.*, 909(1), 1–5. <https://doi.org/10.1088/1742-6596/909/1/012038>
- [20] Pourghanbar, S., Manafian, J., Ranjbar, M., Aliyeva, A., and Gasimov, Y. S. (2020). An efficient alternating direction explicit method for solving a nonlinear partial differential equation. *Math. Probl. Eng.*, 2020(November), 1–12. <https://doi.org/10.1155/2020/9647416>
- [21] Harley, C. (2010). Hopscotch method: The numerical solution of the Frank-Kamenetskii partial differential equation. *Appl. Math. Comput.*, 217, 4065–4075. <https://doi.org/10.1016/j.amc.2010.10.020>
- [22] Al-Bayati, A., Manaa, S., and Al-Rozbayani, A. (2011). Comparison of finite difference solution methods for reaction diffusion system in two dimensions. *AL-Rafidain J. Comput. Sci. Math.*, 8(1), 21–36. <https://doi.org/10.33899/csmj.2011.163605>
- [23] Nwaigwe, C. (2022). An unconditionally stable scheme for two-dimensional convection-diffusion-reaction equations. https://scholar.google.co.uk/citations?view_op=view_citation&hl=en&user=PW9Enr0AAAJ&citation_for_view=PW9Enr0AAAAJ:M3ejUd6NZC8C
- [24] Savović, S., Drljača, B., and Djordjevich, A. (2022). A comparative study of two different finite difference methods for solving advection–diffusion reaction equation for modeling exponential traveling wave in heat and mass transfer processes. *Ric. di Mat.*, 71, 245252. <https://doi.org/10.1007/s11587-021-00665-2>
- [25] Berger, J., Gasparin, S., Mazuroski, W., and Mendes, N. (2020). An efficient two-dimensional heat transfer model for building envelopes. *Numer. Heat Transf. Part A Appl.*, 79(3), 163–194. <https://doi.org/10.1080/10407782.2020.1836936>
- [26] Kovács, E., and Gilicz, A. (2018). New stable method to solve heat conduction problems in extremely large systems. *Des. Mach. Struct.*, 8(2), 30–38.
- [27] Kovács, E. (2020a). New stable, explicit, first order method to solve the heat conduction equation. *J. Comput. Appl. Mech.*, 15(1), 3–13. <https://doi.org/10.32973/jcam.2020.001>
- [28] Saleh, M., Nagy, Á., and Kovács, E. (2020a). Part 2: Construction and investigation of new numerical algorithms for the heat equation. *Multidiszcip. tudományok*, 10(4), 339–348. <https://doi.org/10.35925/j.multi.2020.4.37>
- [29] Saleh, M., Nagy, Á., and Kovács, E. (2020b). Part 3: Construction and investigation of new numerical algorithms for the heat equation. *Multidiszcip. tudományok*, 10(4), 349–360. <https://doi.org/10.35925/j.multi.2020.4.38>
- [30] Nagy, Á., Saleh, M., Omle, I., Kareem, H., and Kovács, E. (2021a). New stable, explicit, shifted-hopscotch algorithms for the heat equation. *Math. Comput. Appl.*, 26(3), 61, <https://doi.org/10.3390/mca26030061>
- [31] Nagy Á., Omle, I., Kareem, H., Kovács, E., Barna, I. F., and Bogнар, G. (2021b). Stable, explicit, Leapfrog-Hopscotch algorithms for the diffusion equation. *Computation*, 9(8), 92. <https://doi.org/10.3390/computation9080092>
- [32] Kovács, E., Nagy, Á., and Saleh, M. (2021). A set of new stable, explicit, second order schemes for the non-stationary heat conduction equation. *Mathematics*, 9(18), 2284. <https://doi.org/10.3390/math9182284>
- [33] Jalghaf, H. K., Kovács, E., Majár, J., Nagy, Á., and Askar, A. H. (2021). Explicit stable finite difference methods for diffusion-reaction type equations. *Mathematics*, 9(24), 3308. <https://doi.org/10.3390/math9243308>

- [34] Kovács, E. (2020b). A class of new stable, explicit methods to solve the non-stationary heat equation. *Numer. Methods Partial Differ. Equ.*, 37(3), 2469–2489. <https://doi.org/10.1002/num.22730>
- [35] Kovács, E., Nagy, Á., and Saleh, M. (2022). A new stable, explicit, third-order method for diffusion-type problems. *Adv. Theory Simulations*, 5(6), 2100600. <https://doi.org/10.1002/adts.202100600>
- [36] Saleh, M., Kovács, E., Barna, I. F., and Mátyás, L. (2022). New analytical results and comparison of 14 numerical schemes for the diffusion equation with space-dependent diffusion coefficient. *Mathematics*, 10(15), 2813. <https://doi.org/10.3390/math10152813>
- [37] Munka, M. and Pápay, J. (2001). *4D numerical modeling of petroleum reservoir recovery*. Budapest: Akadémiai Kiadó.
- [38] Gourlay, A. R. and McGuire, G. R. (1971). General hopscotch algorithm for the numerical solution of partial differential equations. *IMA J. Appl. Math.*, 7(2), 216–227. <https://doi.org/10.1093/imamat/7.2.216>
- [39] Saleh, M. and Kovács, E.: *New explicit asymmetric hopscotch methods for the heat conduction equation*. 2021 The 1st International Electronic Conference on Algorithms, vol. 2, no. 1, p. 22, <https://doi.org/10.3390/IOCA2021-10902>
- [40] Hirsch, C. (1988). *Numerical computation of internal and external flows, volume 1: Fundamentals of numerical discretization*. Wiley.
- [41] Chapra, S. C. and Canale, R. P. (2015). *Numerical methods for engineers*. Seventh Edition, 7th ed. New York: McGraw-Hill Science/Engineering/Math.
- [42] Iserles, A. (2009). *A first course in the numerical analysis of differential equations*. Cambridge: Cambridge Univ. Press. <https://doi.org/10.1017/CBO9780511995569>
- [43] Olver, F. W. J., Lozier, D. W., Boisvert, R. F., and Clark, C. W. (2011). *NIST handbook of mathematical functions*. vol. 66, no. 1. New York: Cambridge Univ. Press.
- [44] Wikipedia. *Whittaker function*. Available: https://en.wikipedia.org/wiki/Whittaker_function