Feladatok a 6. hét anyagához (beadható: a 7. heti gyakorlatig)

Programírás esetén a Matlab fájlokat kell elküldeni részemre e-mailben. A fájlok ne függvények, hanem szkriptek legyenek, azaz olyan m-fájlok, amik beavatkozás nélkül maguktól lefutnak. A nem programozási feladatokat lapon (kézzel írva vagy nyomtatva) kell beadni.

1. Feladat. (Matlab) Oldjuk meg a

$$\begin{aligned} \frac{\partial u}{\partial t}+\frac{\partial u}{\partial x} &= 0,\ x\in(0,1),\ t\in(0,4),\\ u_0(0,x) &= 1+\sin(2\pi x),\\ u(t,0) &= u(t,1) \end{aligned}$$

advekciós egyenletet a Crank–Nicolson-sémával! Alkalmazzuk a Sherman–Morrison-formulát az iterációs egyenletrendszer megoldására (lásd az előadás diáit)! Legyen $r=9$ és $n=300$! A program a szimulációt mutassa!

2. Feladat. (Matlab) Oldjuk meg a

$$\begin{aligned} \frac{\partial u}{\partial t}+\frac{\partial u}{\partial x} &= 0,\ x\in(0,1),\ t\in(0,5.3),\\ u_0(0,x) &= 1+\sin(2\pi x),\\ u(t,0) &= u(t,1) \end{aligned}$$

advekciós egyenletet a leap-frog-séma segítségével (lásd az előadás diái)! Ez egy háromréteges séma, így nem elegendő a 0. időréteg a séma indításához, hanem kell egy közelítés az 1. időrétegen is! Használjuk az 1. időréteges közelítéshez a nem stabil FTCS-sémát (lásd előadás diái). Meg fog-e így maradni a leap-frog-séma $\mathcal{O}(\Delta t^2+\Delta x^2)$ rendje? Ennek kiderítésére végezzünk numerikus kísérleteket az $r=0.3$ választás mellett az $n=50,100,200$ választásokkal! A program eredményeként adjon egy 3×2-es mátrixot, melynek első oszlopa a három esetben használt rácstávolság ($\Delta x=1/n$), a második oszlopa pedig a megfelelő rácstávolsághoz tartozó maximumnormabeli hiba! Értékeljük az eredményt!

3. Feladat. (Matlab) Oldjuk meg a

$$\begin{aligned} \frac{\partial u}{\partial t}+\frac{\partial u}{\partial x} &= 0,\ x\in(0,1),\ t\in(0,1.2),\\ u_0(0,x) &= 1+\sin(2\pi x),\\ u(t,0) &= g(t)\equiv 1 \end{aligned}$$

Dirichlet-peremfeltételes advekciós egyenletet a Lax–Friedrichs-séma segítségével ($u_j^{k+1}=(1+r)u_{j-1}^k/2+(1-r)u_{j+1}^k/2$)! A jobb oldalon használjuk az $u_n^{k+1}=u_n^k-r(u_n^k-u_{n-1}^k)$ numerikus peremfeltétel! Erre azért van szükség, mert a differenciacsillag alakja miatt a séma nem tudja kiszámolni az $x_n=1$ pontbeli közelítéseket. Ezt a hiányt nekünk kell pótolni a programírás során. Az iteráció tehát csak az $x_1,\dots,x_{n-1}$ pontokbeli közelítéseket fogja meghatározni. Az $x_0=0$-beli érték a peremfeltételből következik, az x_n-beli értéket pedig a numerikus peremfeltétel adja. A programnak a szimulációt kell mutatnia $r=0.9$, $n=100$ választással.