

Párhuzamosított idődiszkretizáló módszerek és alkalmazásuk mélytanulási feladatokban

Kupás Vendel Péter

Témavezető

Dr. Fekete Imre

Eötvös Loránd Tudományegyetem

2022. december 22.

- 1 Bevezetés
- 2 Időbeli többrácsos redukciós módszer (MGRIT)
 - Egylépéses módszerek
 - Többrácsos módszerek
 - MGRIT
 - Nemlineáris eset
- 3 Neurális közönséges differenciálegyenletek
- 4 Kísérletek és további tervek

- Neurális közönséges differenciálegyenlet
 - A reziduális hálók folytonos megfelelői
 - Tanítás: Adjungált érzékenység (adjoint sensitivity) módszer
 - Kezdetiérték-feladat

- Időbeli többrácsos redukciós módszer (MGRIT)
 - Időben párhuzamosítható numerikus módszer kezdetiérték-feladatok megoldására
 - Parareal algoritmus
 - Iteratív eljárás

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

$$u(0) = u_0$$

- Legyen $0 = t_0 < t_1 < \dots < t_{N-1} < t_N = T$ a $[0, T]$ intervallum δt lépésközű felosztása.

$$u(0) = u_0$$

$$u_i = \Phi_i(u_{i-1}) + g_i, \quad i = 1, 2, \dots, N$$

- Ha f lineáris és $\Phi_i(u_{i-1}) = \Phi u_{i-1}$ ($i = 1, \dots, N$), akkor a módszer $Au = g$ alakba írható, ahol

$$A = \begin{bmatrix} I & & & & \\ -\Phi & I & & & \\ & & \ddots & & \\ & & & \ddots & \\ & & & & -\Phi & I \end{bmatrix}$$

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

$$u(0) = u_0$$

- Legyen $0 = t_0 < t_1 < \dots < t_{N-1} < t_N = T$ a $[0, T]$ intervallum δt lépésközű felosztása.

$$u(0) = u_0$$

$$u_i = \Phi_i(u_{i-1}) + g_i, \quad i = 1, 2, \dots, N$$

- Ha f lineáris és $\Phi_i(u_{i-1}) = \Phi u_{i-1}$ ($i = 1, \dots, N$), akkor a módszer $Au = g$ alakba írható, ahol

$$A = \begin{bmatrix} I & & & & & \\ -\Phi & I & & & & \\ & & \ddots & & & \\ & & & \ddots & & \\ & & & & -\Phi & I \end{bmatrix}$$

- Iteratív módszer
- Klasszikus elliptikus parciális differenciálegyenletek.
- Minden Ω_l , $l = 1, \dots, L$ rácshoz tartozik egy $A_l u = g_l$ lineáris egyenletrendszer.

1. Algorithm 2-rácsos MG az $A_h u_h = g_h$ megoldására

- 1: Elősimítás a finom hálón $u_h = S(u_h, g_h)$
 - 2: Reziduális számítása: $r_h = g_h - A_h u_h$
 - 3: Leszűkítés: $r_H = R r_h$
 - 4: Az $A_H e_H = r_H$ megoldása
 - 5: Prolongálás: $u_h = u_h + P e_H$
 - 6: Utósimítás a finom hálón: $u_h = S(u_h, g_h)$
-

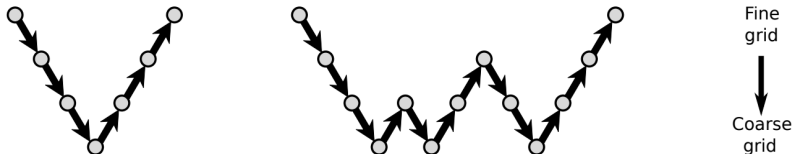
- Kettőnél több rács esetén a 4. pontban lévő egyenletrendszert nem direkt, hanem rekurzívan oldjuk meg.



MG V-ciklus (bal) és F-ciklus (jobb)

- Tér-idő MG: időfüggő parciális differenciálegyenlet

- Kettőnél több rács esetén a 4. pontban lévő egyenletrendszert nem direkt, hanem rekurzívan oldjuk meg.



MG V-ciklus (bal) és F-ciklus (jobb)

- Tér-idő MG: időfüggő parciális differenciálegyenlet

- Parareal algoritmus
- A durva rácshoz tartozó lineáris egyenletrendszer mérete még mindig túl nagy lehet
- Rekurzív hívás egy durvább rácson \rightarrow MGRIT (F-relaxálás, V-ciklus)
- Sok változtatható paraméter
 - Egylépéses módszer
 - FCF-relaxáció
 - Ritkítés
 - Rekurzív hívások sorrendje
 - Konvergenciát gyorsító plusz eljárások
 - Implicit módszer egylépéses módszer esetén további paraméterek

- Parareal algoritmus
- A durva rácshoz tartozó lineáris egyenletrendszer mérete még mindig túl nagy lehet
- Rekurzív hívás egy durvább rácson \rightarrow MGRIT (F-relaxálás, V-ciklus)
- Sok változtatható paraméter
 - Egylépéses módszer
 - FCF-relaxáció
 - Ritkítés
 - Rekurzív hívások sorrendje
 - Konvergenciát gyorsító plusz eljárások
 - Implicit módszer egylépéses módszer esetén további paraméterek

$$\begin{aligned} \partial_t u(t, x) - \kappa \Delta u(t, x) &= b(t, x), & x \in \Omega &= [0, \pi]^d, & t \in (0, T], & \kappa > 0 \\ u(x, 0) &= u_0(x), & x &\in \Omega \\ u(t, x) &= 0, & x &\in \partial\Omega, & t &\in [0, T] \end{aligned}$$

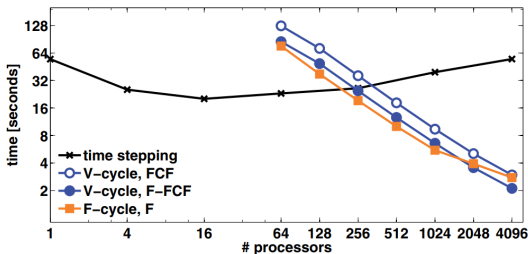


FIG. 9. Time to solve Implicit2D($T = \pi^2$) on a $129^2 \times 16,385$ space-time grid using sequential time stepping and three MGRIT variants.

1. ábra. A futási idő a processzorszám tekintetében az egyes eljárásokra nézve [2].

- Teljes approximációs séma (Full Approximation Scheme, rövidítve FAS)
- Implicit egylépéses módszer $\rightarrow \Phi_l$ kiszámítása \equiv nemlineáris egyenletrendszer megoldása \rightarrow Newton-módszer
- Durva rácsokon nagy lépésköz \rightarrow rossz kezdőérték
- Megoldás: térbeli ritkítás
- Vegyük az Ω_l , $l = 0, 1, \dots, L$ időbeli felosztásokat a δt , $m\delta t$, stb lépésközzel és minden felosztáson használjunk egy egylépéses módszert. Jelölje $A_l u^{(l)} = g^{(l)}$ az Ω_l -hez tartozó egyenletrendszert, ahol A_l mátrixot a Φ_l mátrix határozza meg. Továbbá legyen R_l illetve P_l az időbeli és R_x illetve P_x a térbeli leszűkítő illetve prolongáló operátor.

- Teljes approximációs séma (Full Approximation Scheme, rövidítve FAS)
- Implicit egylépéses módszer $\rightarrow \Phi_I$ kiszámítása \equiv nemlineáris egyenletrendszer megoldása \rightarrow Newton-módszer
- Durva rácsokon nagy lépésköz \rightarrow rossz kezdőérték
- Megoldás: térbeli ritkítás
- Vegyük az Ω_I , $I = 0, 1, \dots, L$ időbeli felosztásokat a δt , $m\delta t$, stb lépésközzel és minden felosztáson használjunk egy egylépéses módszert. Jelölje $A_I u^{(I)} = g^{(I)}$ az Ω_I -hez tartozó egyenletrendszert, ahol A_I mátrixot a Φ_I mátrix határozza meg. Továbbá legyen R_I illetve P_I az időbeli és R_x illetve P_x a térbeli leszűkítő illetve prolongáló operátor.

2. Algorithm MGRIT-FAS(l)

- 1: **if** l a legdurvább szint **then**
- 2: Oldjuk meg az $A_L(u^{(L)}) = A_L(R_l(u^{(L-1)})) + g^{(L)}$ rendszert
- 3: **else**
- 4: Az $A_l(u^{(l)}) = g^{(l)}$ rendszer relaxálása FCF-relaxációval
- 5: Reziduális számítása és szűkítése injekcióval: $g^{(l+1)} = R_l(g^{(l)} - A_l(u^{(l)}))$
- 6: A megoldás leszűkítése: $u^{(l+1)} = R_l(u^{(l)})$
- 7: **if** Térbeli ritkítás **then**
- 8: $u_i^{(l+1)} = R_x(u_i^{(l+1)})$ és $g_i^{(l+1)} = R_x(g^{(l+1)})_i$
- 9: **end if**
- 10: Megoldás a következő szinten: $MGRIT(l + 1)$.
- 11: $e^{(l+1)} = u^{(l+1)} - u^{(l)}$
- 12: **if** Térbeli ritkítás **then**
- 13: $e_i^{(l+1)} = P_x(e_i^{(l+1)})$
- 14: **end if**
- 15: $u^{(l)} = u^{(l)} + P e^{(l+1)}$
- 16: Az $A_l(u^{(l)}) = g^{(l)}$ rendszer relaxálása F-relaxációval
- 17: **end if**

- Térbeli ritkítás
- Javított kezdőérték használata a Newton-módszernél
- Skip
- Első 3 iteráció alatt nagyobb tolerancia
- Legdurvább rács mérete
- Neumann határfeltételű p -dimenziós Laplace feladat

$$\begin{aligned}u_t(x, t) - \nabla \cdot (|\nabla u(x, t)|^{p-2} \nabla u(x, t)) &= b(x, t), \quad x \in \Omega, \quad t \in [0, T] \\ |\nabla u(x, t)|^{p-2} \nabla u(x, t) \cdot n &= g(x, t), \quad x \in \partial\Omega, \quad t \in [0, T] \\ u(x, 0) &= u_0(x), \quad x \in \Omega\end{aligned}$$

- Térbeli ritkítás
- Javított kezdőérték használata a Newton-módszernél
- Skip
- Első 3 iteráció alatt nagyobb tolerancia
- Legdurvább rács mérete
- Neumann határfeltételű p -dimenziós Laplace feladat

$$\begin{aligned}u_t(x, t) - \nabla \cdot (|\nabla u(x, t)|^{p-2} \nabla u(x, t)) &= b(x, t), \quad x \in \Omega, \quad t \in [0, T] \\ |\nabla u(x, t)|^{p-2} \nabla u(x, t) \cdot n &= g(x, t), \quad x \in \partial\Omega, \quad t \in [0, T] \\ u(x, 0) &= u_0(x), \quad x \in \Omega\end{aligned}$$

Solver ID	Spatial grids	IIG	Skip	Cheap first 3 iters	Coarsest grid size	Memory mult., $p = 128 \rightarrow 4096$	Runtime per iter	Iterations	Runtime
0	1	Never	No	No	4	40 \rightarrow 16	162s	7	1135s
1	4	Never	No	No	4	31 \rightarrow 7	101s	7	712s
2	4*	Never	No	No	4	12 \rightarrow 2	57s	30	1717s
3	1	C points	No	No	4	53 \rightarrow 21	91s	7	638s
4	4	C points	No	No	4	42 \rightarrow 10	82s	7	579s
5	1	Always	No	No	4	84 \rightarrow 21	92s	7	647s
6	4	Always	No	No	4	73 \rightarrow 10	84s	7	591s
7	1	C points	Yes	No	4	53 \rightarrow 21	64s	7	453s
8	4	C points	Yes	No	4	42 \rightarrow 10	58s	7	410s
9	1	Always	Yes	No	4	84 \rightarrow 21	61s	7	429s
10	4	Always	Yes	No	4	73 \rightarrow 10	55s	7	391s
11	1	Always	Yes	Yes	4	84 \rightarrow 21	56s	7	395s
12	4	Always	Yes	Yes	4	73 \rightarrow 10	51s	7	360s
13	1	Always	Yes	Yes	16	80 \rightarrow 17	58s	7	408s
14	4	Always	Yes	Yes	16	73 \rightarrow 10	50s	7	354s
15	1	Always	Yes	Yes	64	76 \rightarrow 13	63s	8	506s
16	4	Always	Yes	Yes	64	73 \rightarrow 10	47s	8	383s

Table 4: Overall runtimes, storage costs, iteration counts and average time per iteration for the various solver options, with a $(64)^2 \times 4096$ space-time grid. A * indicates no delay in spatial coarsening.

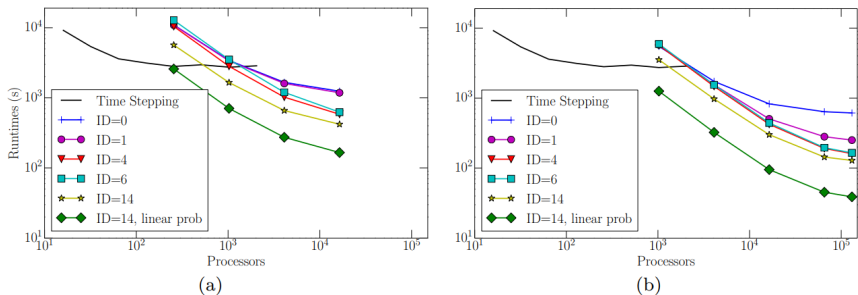


Fig. 5: Strong scaling study for a $(128)^2 \times 16385$ space-time grid, Left: 16 processors in space, $m = 16$, Right: 32 processors in space, $m = 4$.

2. ábra. Az MGRIT futási ideje a modell feladaton [4].

- Vegyünk egy parametrizált kezdetiérték-problémát

$$u'(t) = f_{\theta}(t, u(t))$$

$$u(0) = u_0$$

- Bemenet: $u(0)$
- Output: $u(T)$
- Tanítás: adjungált érzékenység módszer

Adjungált érzékenység módszer (adjoint sensitivity method)

- Tekintsük a következő skalárértékű veszteségfüggvényt:

$$L(u(T)) = L(u(0)) + \int_0^T f_\theta(t, u(t)) dt = L(\text{ODESolve}(u_0, f, 0, T, \theta))$$

- cél: $\frac{dL}{d\theta}$ meghatározása
- Legyen $a(t) = \frac{\partial L}{\partial z(t)}$ az ún adjungált függvény
- Ekkor $a(T)$ ismert és

$$\frac{da(t)}{dt} = -a(t)^T \frac{\partial f_\theta(t, u(t))}{\partial z}$$

- Végül

$$\frac{dL}{d\theta} = - \int_0^T a(t)^T \frac{\partial f_\theta(t, u(t))}{\partial \theta} dt$$

3. Algorithm Reverse-mode derivative of an ODE initial value problem

- 1: **Input:** $\theta, 0, T, u_0, \frac{dL}{du(T)}$
- 2: $\frac{dL}{dT} = \frac{dL}{du(T)} f_\theta(T, u(T))$ ▷ Gradiens T szerint
- 3: $s_0 = [u(T), \frac{dL}{du(T)}, 0_{|\theta|}, -\frac{dL}{dT}]$ ▷ Kezdeti értékek definiálása
- 4: **def** *aug_dynamics*($[u(t), a(t), \cdot], t, \theta$): ▷ Dinamika definiálása
- 5: **return** $[f_\theta(u(t)), t, -a(t)^T \frac{\partial f}{\partial u}, -a(t)^T \frac{\partial f}{\partial \theta}]$ ▷ Vektor-Jacobi számolás
- 6: $[u(0), \frac{\partial L}{\partial u(0)}, \frac{\partial L}{\partial \theta}] = \text{ODESolve}(s_0, \text{aug_dynamics}, T, 0, \theta)$ ▷ Vissza idejű megoldás
- 7: **return** $\frac{\partial L}{\partial u(0)}, \frac{\partial L}{\partial \theta}$ ▷ Gradiensek

- Tehát egy numerikus módszerrel (ODESolve) visszafele tudjuk meghatározni a $\frac{\partial L}{\partial \theta}$ gradienst.
- MGRIT → párhuzamos tanítás

- Eddigi futtatások
 - XBraid [6]
 - Tesztfeladatok
- További tervek
 - Az 1. ábra és a 2. ábra rekonstruálása
 - XBraid kipróbálása neurális közönsége differenciálegyenletek tanításában

-  R. T. Chen, Y. Rubanova, J. Bettencourt, D. Duvenaud: *Neural ordinary differential equations*, Advances in Neural Information Processing Systems 31 (NeurIPS 2018), pp. 6571-6583, 2018
-  R. D. Falgout, S. Friedhoff, Tz. V. Kolev, S. P. MacLachlan, and J. B. Schroder: *Parallel Time Integration with Multigrid*, SIAM J. Sci. Comput., 36 (2014), pp.C635-C661. LLNL-JRNL-645325.
-  Horváth Róbert, Izsák Ferenc, Karátson Kános: *Parciális differenciálegyenletek numerikus módszerei számítógépes alkalmazásokkal*, Elektronikus jegyzet, 2013
-  R. D. Falgout, T. A. Manteuffel, B. O'Neill, and J. B. Schroder: *Multigrid Reduction in Time for Nonlinear Parabolic Problems: A Case Study*, SIAM J. Sci. Comput., 39 (2017), pp 298-322. LLNL-JRNL-692258
-  J.-L. Lions, Y. Maday, and G. Turnici: *A parareal in time discretization of PDEs*, C.R. Acad. Sci. Paris, Serie I, 332 (2001), pp. 661-668.
-  XBraid: Parallel multigrid in time. <http://lnl.gov/casc/xbraid>