# Párhuzamosított idődiszkretizáló módszerek

Kupás Vendel Péter

2021.12.09.

A mai processzorok elérték a maximális teljesítőképességük határát, így a többmagos számítógépek megjelenése óta természetes igény differenciálegyenletek párhuzamos megoldási módszereinek kidolgozása.
Ha adott egy kezdetiérték probléma az $[a, b]$ időintervallumon, akkor az idődiszkretizálás alatt azt értjük, hogy felosztjuk az intervallumot $N$ részre az $a = t_0 < t_1 < \ldots < t_{N-1} < < t_N = b$ pontokkal és az $y_0, y_1, \ldots, y_N$ értékekkel közelítjük a megoldás $t_0, t_1, \ldots, t_N$ helyen felvett értékeit. Az $y_0, y_1, \ldots, y_N$ mennyiségek kiszámolását a numerikus módszerünk határozza meg. Ilyen jól ismert szekvenciális módszerek például a Runge–Kutta és a többlépéses módszerek.
A félév során Martin Gander összefoglaló cikkében [1] szereplő módszereket ismertem meg.

## 1. Osztályozás

A párhuzamosított módszereket 3 nagyobb osztályba lehet sorolni aszerint, hogy milyen értékeket számítunk ki egyszerre.
**Párhuzamosítás a feladaton keresztül** Itt olyan módszerek szerepelnek, ahol a problémát több kisebb részproblémára bontjuk a komponensek szerint és ezeket oldjuk meg párhuzamosan. Ebben az osztályban nem szüséges idődiszkretizációt végrehajtani. A legismertebb módszer ebből az osztályból a hullámforma relaxáció (waveform relaxation). Itt definiálunk egy iterációt, melynek minden lépésében skalár differenciálegyenleteket oldunk meg párhuzamosan.
**Párhuzamosítás a lépéseken keresztül** Itt az időintervallumnak vesszük egy felosztását és a módszer egy lépésében az időtengely mentén több pontban végzünk számításokat párhuzamosan.
**Párhuzamosítás a módszeren keresztül** Ebbe az osztályba nagyon sokféle módszer tartozik. Minden olyan módszer ide sorolható, ahol a szükséges számításokat párhuzamosan tudjuk végezni. Például ha egy olyan idődiszkretizáló Runge–Kutta módszerünk van, ahol $k_i$ és $k_{i+1}$ magfüggvények is csak a korábban számolt lépcsőktől függnek, akkor tudjuk őket egyszerre számolni. A direkt módszerek nagy része is ide sorolható.

## 2. Belövéses módszerek

Ebben a félévben ezekkel a módszerekkel foglalkoztam részletesebben. Ezek a módszerek a második osztályba tartoznak.
Egy differenciálegyenlet megoldása a $t$ időpontban befolyásolja, hogy a későbbiekben miként fog viselkedni, ezért az időtengely mentén történő párhuzamosítás nem magától értetődő, hogyan valósítható meg.

Az alapötlet az, hogy egy kezdetiérték probléma megoldása az egész intervallumon előáll úgy, mint az osztópontok által meghatározott részintervallumokon vett kezdetiérték problémák megoldásainak az egyesítése, ahol az $y_i$ kezdetiérték a $[t_{i-1}, t_i]$ intervallumon vett megoldás $t_i$ helyen felvett értéke.
Ha ezen feltételeket felírjuk és a részintervallumokon a pontos megoldást feltételezzük, vagy egy szekvenciális numerikus módszerrel közelítjük, akkor az $y_0, y_1, \ldots, y_N$ változókra egy nemlineáris egyenletrendszert kapunk, melynek megoldására alkalmazhatjuk a Newton-módszert, vagy hozzá hasonló iterációs eljárásokat.
A módszer továbbra is szekvenciális marad, de a frissítéshez szükséges költséges értékeket tudjuk párhuzamosan számítani. Akkor érhetünk el gyorsítást, ha $N$-nél kevesebb iterációt kell végrehajtanunk a megoldás jó közelítéséhez, valamint sok processzor áll a rendelkezésünkre.
Ha a Newton-módszert használjuk, akkor a módszer nem feltétlenül hatékony minden jobboldal esetén, de létezik egy nagy függvényosztály, amely esetén ha a részintervallumokon a pontos megoldást elő tudjuk állítani és mindent pontosan ki tudunk számítani, akkor a konvergencia lokálisan kvadratikus lesz. Ha pedig egy megfelelő numerikus módszert használunk, akkor szuperlineáris. [2]

Ebből a családból Bellen és Zennaro algoritmusát [3] valamint a parareal módszereket [4] ismertem meg és teszteltem. Mind a kettő a Newton-módszernek egy olyan változatát használja az iteráció során, ahol nem szükséges deriváltakat számolni. Olyan példákon próbáltam ki őket, ahol ismert a megoldás. Ezekből azt kaptam, hogy az osztópontok számánál jóval kevesebb iteráció elegendő a megoldás jó közelítéséhez. Tehát ha sok processzor áll rendelkezésre, akkor jó implementálás esetén valóban gyorsítást lehet elérni.

## Hivatkozások


[1] M. GANDER, 50 Years of Time Parallel Time Integration, Multiple Shooting and Time Domain Decomposition Methods. Contributions in Mathematical and Computational Sciences, vol 9. Springer, Cham, pp. 69-113, 2015

[2] P. CHARTIER AND B. PHILIPPE, A parallel shooting technique for solving dissipative ODEs, Computing, 51 (1993), pp. 209–236.

[3] A. BELLEN AND M. ZENNARO, Parallel algorithms for initial-value problems for difference and differential equations, J. Comput. Appl. Math., 25 (1989), pp. 341–350.

[4] J.-L. LIONS, Y. MADAY, AND G. TURINICI, A parareal in time discretization of PDEs, C.R. Acad. Sci. Paris, Serie I, 332 (2001), pp. 661–668.