

5. Markov populációs folyamatok, terheléselosztás

Kommunikációs hálózatok teljesítményének elemzése

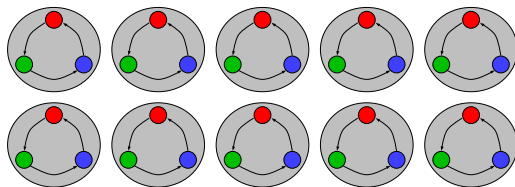
Horváth Illés

2024/10/02

- (1) Markov populációs folyamatok
- (2) Példa: terheléselosztás
- (3) Mean-field konvergencia
- (4) Numerikus diffegyenlet-megoldás

Markov populációs folyamatok

Egy folytonos idejű Markov-láncból vegyünk N példányt, amik párhuzamosan zajlanak. Egyetlen mini Markov-lánc állapottere K méretű, generátora $Q = (r_{ij})_{i,j=1}^K$.



Markov populációs folyamatok

Hogy lehet egy ilyen rendszer pillanatnyi állapotát leírni?

Egy lehetőség, hogy minden egyes Markov-lánc állapotát nyilvántartjuk („teljes rendszer”). Ekkor az állapottér mérete K^N .

Egy másik lehetőség, hogy csak azt tartjuk számon, hogy az egyes állapotokban mennyien tartózkodnak: jelölje $X_k(t)$ a k állapotban tartózkodó Markov-láncok számát t -kor. Ekkor az állapottér

$$\left\{ (X_1, \dots, X_K) : X_k \in \mathbb{Z}, X_k \geq 0, \sum_{k=1}^K X_k = N \right\},$$

aminek a mérete N^K nagyságrendű. Ez is nagy (bár kisebb, mint a teljes rendszer), viszont jól strukturált. Ezt fogjuk használni.

Markov populációs folyamatok

A legegyszerűbb esetben az egyes mini Markov-láncok függetlenül zajlanak. Ilyenkor a Markov-tulajdonság teljesül a teljes rendszerre is, és $X(t)$ -re is.

Példa. Legyen $N = 10$ és $K = 3$. $X(t)$ -nek az $(5, 3, 2)$ állapotból a lehetséges átmenetei:

- ▶ $(5, 3, 2) \rightarrow (4, 4, 2)$, ennek rátája $5r_{12}$,
- ▶ $(5, 3, 2) \rightarrow (4, 3, 3)$, ennek rátája $5r_{13}$,
- ▶ $(5, 3, 2) \rightarrow (6, 2, 2)$, ennek rátája $3r_{21}$,
- ▶ $(5, 3, 2) \rightarrow (5, 2, 3)$, ennek rátája $3r_{23}$,
- ▶ $(5, 3, 2) \rightarrow (6, 3, 1)$, ennek rátája $2r_{31}$,
- ▶ $(5, 3, 2) \rightarrow (5, 4, 1)$, ennek rátája $2r_{32}$.

Markov populációs folyamatok

Azt szeretnénk, hogy az egyes Markov-láncok ne legyenek függetlenek. Például megengedhetjük, hogy az átmenetráták függjenek a többi Markov-lánc állapotától.

Ha teljesül, hogy az r_{ij} ráták az

$$x(t) = \frac{X(t)}{N}$$

függvényei, akkor ezt *sűrűségfüggő Markov populációs folyamatnak* hívjuk. $x(t)$ az $X(t)$ normalizált változata, ami azt mondja meg, hogy a Markov-láncok mekkora része van az egyes állapotokban.

Ilyenkor az egyes Markov-láncok már nem függetlenek. Fontos: a Markov-tulajdonság teljesül $x(t)$ -re!

Példa: terhelés-elosztás

Mi történik, amikor rákattintunk egy YouTube-videóra?

Beküldünk egy igényt, amit majd kiszolgál egy szerver. Na de a YouTube-nak sok szervere van, melyikhez kerül a mi igényünk?

A beérkező igényeket egy külön szerver (“diszpécser”) továbbítja a tényleges kiszolgáló szerverek valamelyikéhez. A cél az igények minél hatékonyabb kiszolgálása, ehhez sok szempontot igénybe vehet, pl.

- ▶ a szerverek leterheltsége
- ▶ az igény típusa
- ▶ fizikai távolság
- ▶ stb.

Terhelés-elosztás

Példa. Mi történik, amikor rákattintunk egy YouTube-videóra?

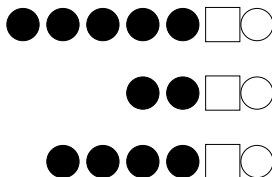
Beküldünk egy igényt, amit majd kiszolgál egy szerver. Na de a YouTube-nak sok szervere van, melyikhez kerül a mi igényünk?

A beérkező igényeket egy külön szerver (“diszpécser”) továbbítja a tényleges kiszolgáló szerverek valamelyikéhez. A cél az igények minél hatékonyabb kiszolgálása, ehhez sok szempontot igénybe vehet, pl.

- ▶ **a szerverek leterheltsége**
- ▶ az igény típusa
- ▶ fizikai távolság
- ▶ stb.

Terhelés-elosztás

Példa. Hasonlóan működnek a kasszák is egy áruházban. Melyik sorba állnánk be a 3 közül?



Nyilván a legrövidebbhez. Ez a JSQ (Join-Shortest-Queue) néven ismert terhelés-elosztási elv.

Példa. Amikor autós navigációs alkalmazások alternatív útvonalakat keresnek, az is terhelés-elosztás.

Erre a helyzetre fogunk matematikai modellt adni.

Terhelés-elosztás

Tekintsük a következő kiszolgáló rendszert. A rendszer N szerverből és egy diszpécserből áll. Az igények mind a diszpécserhez futnak be, amely minden egyes igényt továbbít valamelyik szerverhez valamilyen terhelés-elosztási elv alapján (pl. JSQ).

Minden egyes szervernek van egy sora, amiben a beérkező igényeket tárolja. Az egyszerűség kedvéért tegyük fel, hogy a szerverek egyformák és véges K nagyságú bufferrel rendelkeznek. (Egyébként ha többféle szerver van, az sem gond.) A diszpécsernek nincs sora, azonnal továbbítja a beérkező igényeket.

Feltesszük, hogy a bejövő igények Poisson-folyamat szerint érkeznek λN rátával, valamint a kiszolgálási idő is exponenciális. Ekkor a teljes rendszerre teljesül a Markov-tulajdonság.

Terhelés-elosztás

Egy szerver állapota az, hogy hány igény van a sorában; az egy szerverhez tartozó mini Markov-lánc állapottere $\{0, 1, \dots, K\}$.

Ebben az esetben $X_0(t)$ azt jelöli, hogy a t időpontban hány üres szerver van, $X_1(t)$ azt, hogy a t időpontban hány szerverben van 1 igény, és így tovább, egészen $X_K(t)$ -ig. Az

$$x(t) = \frac{X(t)}{N}$$

vektor pedig a megfelelő terheltségű szerverek aránya.

Kiszolgálás

A kiszolgálás lehet

- ▶ FIFO (First-In-First-Out), mint pl. a bolti sorokban;
- ▶ vagy párhuzamos, PS (Processor Sharing), amikor egy szerver több igényt szolgál ki párhuzamosan.

Feltesszük, hogy minden egyes szervernél a kiszolgálási idő exponenciális. Egy szerver kiszolgálási rátája függhet attól, hogy mennyi a bent lévő igények k száma. Ez főleg PS-nél releváns, amikor egy szerver több igényt hatékonyan tud párhuzamosan kiszolgálni, mint egyet; FIFO esetén a kiszolgálási ráta általában konstans.

Feltesszük, hogy a bejövő igények Poisson-folyamat szerint érkeznek λN rátával. Ekkor a teljes rendszer Markov-folyamat.

Terhelés-elosztás

Példa. Tekintsük a következő paramétereket:

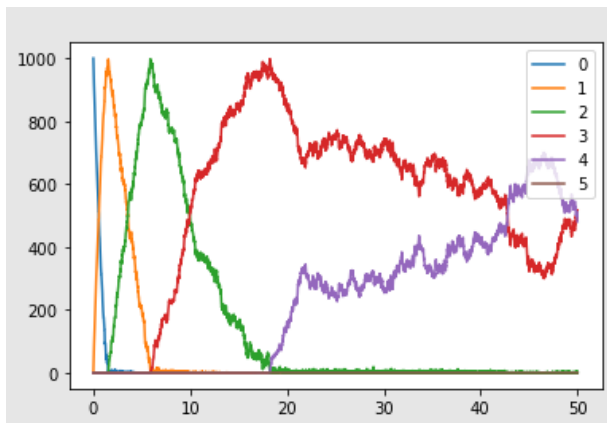
- ▶ $N = 1000$ szerver,
- ▶ egy szerver maximum 5 igényt szolgál ki egyszerre (ha több igény van bent, a többi sorban áll),
- ▶ egy szerver teljes kiszolgálási rátája a szerverben bent lévő igények számának függvényében:

k	1	2	3	4	5
μ_k	1.0	1.1	1.2	1.3	1.4

- ▶ $\lambda = 1.25$; az egész rendszerbe az érkezési ráta λN ,
- ▶ kezdeti feltétel: üres rendszer,
- ▶ JSQ terheléselosztás.

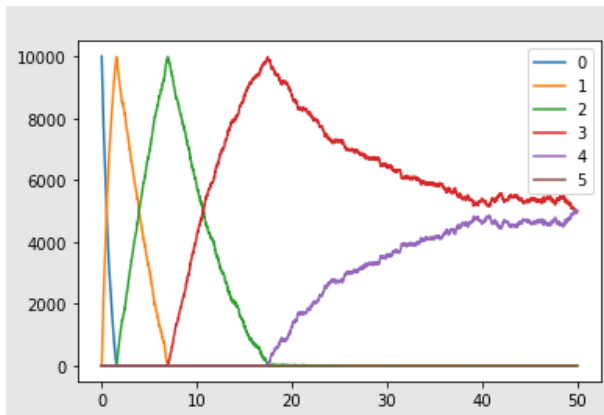
Hogy nézhet ki ennek a rendszer egy realizációja? Gondoljuk meg előre, hogy mit várunk.

Terhelés-elosztás



Terhelés-elosztás

Ugyanez $N = 10000$ szerver esetén:



Mean-field egyenlet-rendszer

A fenti ábra alapján az sejthető, hogy amint $N \rightarrow \infty$, a Markov populációs folyamat koncentrálódik valami determinisztikus folyamatra.

Egy általános sűrűségfüggő Markov populációs folyamathoz tartozó mean-field (térátlag) egyenlet-rendszer a következő:

$$\frac{d}{dt}v_k(t) = \sum_{i \neq k} v_i(t)r_{ik}(v(t)) - \sum_{i \neq k} v_k(t)r_{ki}(v(t)), \quad k = 1, \dots, K$$

Ez egy K -dimenziós közönséges differenciálegyenlet-rendszer. A megoldás általában létezik és egyértelmű.

Kurtz-tétel

Tétel (Kurtz I.)

Tegyük fel, hogy

- ▶ $x(0) \rightarrow v(0)$ valószínűségben, amint $N \rightarrow \infty$.

Ekkor a mean-field egyenletrendszer $v(t)$ megoldása egyértelmű, és tetszőleges véges T -re és $\varepsilon > 0$ -ra

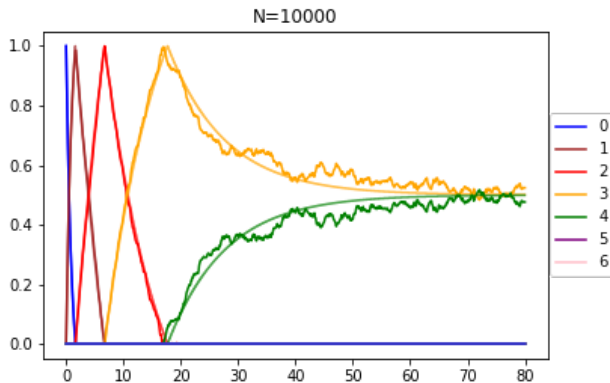
$$\lim_{N \rightarrow \infty} \mathbb{P} \left(\max_{0 \leq t \leq T} \max_{1 \leq k \leq K} |v_k(t) - x_k(t)| > \varepsilon \right) = 0.$$

Nem biz.

$v(t)$ -t hívjuk a populációs folyamat mean-field limeszének is.

Terhelés-elosztás

$N = 10000$ és a mean-field limesz ($N = \infty$):



Kurtz-tétel

Megjegyzések.

Az, hogy $v(t)$ egy közönséges (és nem pl. egy késleltetett) diffegyenlet-rendszer megoldása, annak felel meg, hogy a mean-field limesz memóriamentes, ami éppen azon múlik, hogy az eredeti populációs folyamatra teljesül a Markov-tulajdonság.

Ha az r_{ij} átmenetráták nem folytonos függvények, akkor $v(t)$ -nek lehetnek törései – ez általában nem gond, a Kurtz-tétel ilyenkor is teljesül.

A sűrűségfüggő feltevés annak felel meg, hogy minden mini Markov-lánc x^N -et, vagyis a teljes globális állapotot érzékeli. Ha van a populációnak belső struktúrája (pl. ismeretségi körök), akkor általában nem ez a helyzet. Ezzel együtt közelítésnek még ilyen esetben is lehet használni a Kurtz-tételt.

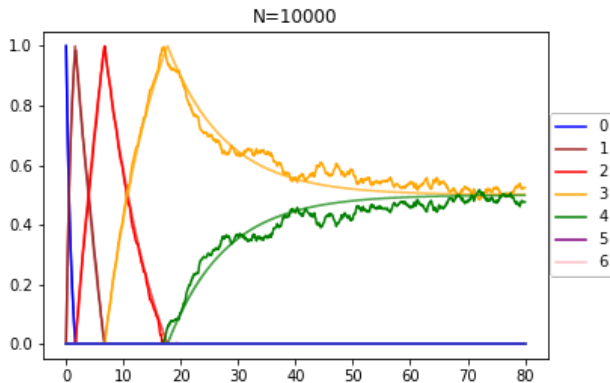
Mean-field limesz

Mire jó a mean-field limesz vizsgálata?

- ▶ nagyméretű rendszerek esetén a teljes rendszer állapottere hatalmas, ehelyett elég egy kisebb méretű diffegyenlet-rendszert vizsgálni, ami numerikusan könnyen megoldható;
- ▶ sok jellemző definiálható és kiszámítható a mean-field rendszerben is: például a mean-field limesz alapján megbecsülhető, hogy a populációs folyamat mikor éri el a stacionárius állapotát.
- ▶ Pl. sorbanállási rendszerek esetén a mean-field limeszben is teljesül a Little-formula, sőt, a kiszolgálási idő teljes eloszlását is ki lehet számítani.

Terhelés-elosztás

$N = 10000$ és a mean-field limesz ($N = \infty$):



Stacionárius eloszlás

Az ábra kétfajta konvergenciát is illusztrál:

- ▶ $N \rightarrow \infty$ esetén a Markov populációs folyamat tart a mean-field limeszhez a $[0, T]$ intervallumon egyenletesen;
- ▶ $t \rightarrow \infty$ esetén a Markov populációs folyamat tart a stacionárius állapotához, a mean-field limesz pedig egy aszimptotikusan stabil fixponthoz (attraktor).

Tétel (Kurtz II.)

Tegyük fel, hogy:

- ▶ *minden N -re $x(t)$ -nek egyetlen stacionárius eloszlása van, π_N , és*
- ▶ *a $v(t)$ -t definiáló diffegyenlet-rendszernek egyetlen $(v_1, \dots, v_K)$ fixpontja van, és az aszimptotikusan stabil.*

Ekkor a π_N eloszlások koncentrálódnak $(v_1, \dots, v_K)$ -ra, amint $N \rightarrow \infty$.

Nem biz.

Diszpécser függvény

Vissza a terhelés-elosztáshoz.

A mean-field diffegyenlet-rendszer felírásához szükségünk lesz a különböző hosszúságú sorokban a kiszolgálási rátára és az érkezési rátára is. Az érkezési ráta a teljes rendszerre λN , de az, hogy ez hogyan terhelődik a különböző hosszúságú sorokra, a terhelés-elosztási elven múlik.

A terhelés-elosztási elvet az ún. diszpécser-függvény segítségével írjuk le. Jelölje $f_k(x)$ annak a valószínűségét, hogy egy érkező igény k hosszú sorba kerül, feltéve, hogy a normalizált globális állapot x .

Pl. JSQ terhelés-elosztási elv esetén:

- ▶ ha van k -nál rövidebb sor, akkor k hosszú sorba nem kerül igény;
- ▶ ha k hosszú a legrövidebb sor, akkor a beérkező igény 1 valószínűséggel k hosszú sorba kerül,

így

$$f_k(x) = \begin{cases} 0 & \text{ha } \sum_{i=0}^{k-1} x_i > 0; \\ 1 & \text{ha } \sum_{i=0}^{k-1} x_i = 0 \text{ és } x_k > 0. \end{cases}$$

Diszpécser függvény

- ▶ Mi a helyzet, ha a beérkező igényt találomra irányítjuk valamelyik sorba (azaz nem történik érdemi terhelés-elosztás)?

Ekkor egyszerűen

$$f_k(x) = x_k.$$

- ▶ JSQ(d) terheléselosztásnál a diszpécser kiválaszt d szerveret találomra, és azok közül a legrövidebbhez küldi az igényt. Ebben az esetben

$$f_k(x) = (x_k + \dots + x_K)^d - (x_{k+1} + \dots + x_K)^d.$$

Az $(f_0(x), \dots, f_K(x))$ függvényeket kollektíven diszpécser-függvénynek nevezzük.

Mean-field limesz

A diszpécser függvény segítségével fel tudjuk írni a mean-field egyenletrendszert:

$$\begin{aligned}\frac{d}{dt}v_k(t) = & \lambda f_{k-1}(v(t))v_{k-1}(t) - \lambda f_k(v(t))v_k(t) + \\ & + \mu_{k+1}v_{k+1}(t) - \mu_k v_k(t) \quad (k = 0, \dots, K);\end{aligned}$$

a jobboldalon az egyes tagok annak felelnek meg, hogy a k hosszú sorok száma megváltozik, ha...

- ▶ érkezés történik $k - 1$ hosszú sorba, vagy
- ▶ érkezés történik k hosszú sorba, vagy
- ▶ kiszolgálás történik $k + 1$ hosszú sorban, vagy
- ▶ kiszolgálás történik k hosszú sorban.

Stabil fixpont

A korábbi példában a stabil fixpont könnyen kiszámítható.

$\lambda = 1.25$ félúton van $\mu_3 = 1.2$ és $\mu_4 = 1.3$ között, így a fixpont

$$v = (0 \ 0 \ 0 \ 0.5 \ 0.5 \ 0).$$

Használjuk a Little-formulát.

Nincs adatvesztés, tehát az effektív érkezési ráta

$$\lambda = \lambda_e = 1.25,$$

az átlagos sorhossz

$$L = 0.5 \cdot 3 + 0.5 \cdot 4 = 3.5,$$

és így az átlagos kiszolgálási idő

$$W = L/\lambda_e = 2.8.$$

Mean-field stacionárius állapot

A π stabil fixpontot mean-field stacionárius állapotnak is hívjuk (mert megfelel annak, hogy a $t \rightarrow \infty$ és $N \rightarrow \infty$ limeszt is vesszük). Hogyan lehet kiszámítani általában?

1. módszer: megoldjuk numerikusan a mean-field differenciálegyenlet-rendszert, majd veszünk egy nagy t értéket. (Mivel a megoldás $t \rightarrow \infty$ esetén a mean-field stacionárius állapothoz konvergál, ezért nagy t -re már közel lesz hozzá.)
2. módszer: dinamikus egyensúly egyenletekből.

Dinamikus egyensúly egyenletek

Jelölje π a mean-field stacionárius eloszlást. Ha az $f_k(x)$ diszpécser függvények folytonosak π -ben, akkor a dinamikus egyensúly egyenletek:

$$\lambda f_k(\pi) = \mu_{k+1} \pi_{k+1} \quad k = 0, \dots, K - 1.$$

Ez egy algebrai egyenlet-rendszer $(\pi_0, \dots, \pi_K)$ -ra; attól függően, hogy az f diszpécser függvény és a μ_k kiszolgálási ráta görbe mennyire egyszerű vagy bonyolult, ennek a megoldása is lehet egyszerű vagy bonyolult.

Dinamikus egyensúly egyenletek

Ha a diszpécser függvény nem folytonos π -ben (pl. a JSQ ilyen), akkor a helyzet némileg más. JSQ esetén a stabil fixpontban csak kétféle sorhossz valószínűsége pozitív: i_0 és $i_0 - 1$, ahol i_0 a legkisebb olyan sorhossz, amire

$$\mu_{i_0} \geq \lambda.$$

A korábbi példában $i_0 = 3$ volt.

Mi történik, ha egy szerver befejezi az igény kiszolgálását?

Ha ez egy i_0 hosszú sorban történt, akkor a sor hossza $i_0 - 1$ lesz, de ezen kívül semmi különleges nem történik.

Ha viszont a kiszolgálás egy $i_0 - 1$ hosszú sorban történt, akkor a sor hossza $i_0 - 2$ -re változik, ami rövidebb, mint az összes többi sor, és JSQ miatt ilyenkor a következő igényt fixen ebbe a sorba irányítja a diszpécser. Viszont a mean-field limeszben az igények végtelen sűrűséggel érkeznek $\rightarrow$ ez a sor azonnal visszatöltődik $i_0 - 1$ hosszúságra.

Dinamikus egyensúly egyenletek

Összességében $i_0 - 2$ hosszú sort nem látunk pozitív ideig, tehát a mean field stacionárius eloszlásban továbbra is csak két sorhossz valószínűsége pozitív, $i_0 - 1$ és i_0 hossza.

De ettől még az igények pozitív arányú része fog ilyen 'instant diszpécseles' keretében egy olyan sorba kerülni, ami az adott pillanatban rövidebb volt, mint az összes többi.

Az $i_0 - 1$ hosszú sorokban a kiszolgálási ráta összesen

$$\lambda_u = \pi_{i_0-1} \mu_{i_0-1},$$

emiatt a teljes bejövő λ érkezési ráta ekkora része fog arra fordítódni, hogy az $i_0 - 1$ hosszú sorokban a kiszolgálást ellensúlyozza, és visszatöltse őket instant $i_0 - 1$ hosszúságra (az u betű az "upkeep" rövidítése).

Dinamikus egyensúly egyenletek

A maradék $\lambda - \lambda_u$ érkezési ráta az $i_0 - 1$ sorokat fogja i_0 hosszúságúra tölteni; a dinamikus egyensúly egyenlet szerint ez egyensúlyban van az i_0 hosszú sorokban történő kiszolgálással. Ennek megfelelően JSQ esetén a dinamikus egyensúly egyenlet a következő alakú:

$$(\lambda - \lambda_u)f_{i_0-1}(\pi) = \mu_{i_0}\pi_{i_0}.$$

Ez nemcsak JSQ esetén van így; általában akkor van upkeep, ha a diszpécser függvény nem folytonos a π mean field stacionárius állapotban. A dinamikus egyensúly egyenletek általános alakja ilyenkor

$$(\lambda - \lambda_u)f_k(\pi) = \mu_{k+1}\pi_{k+1} \quad k = 0, \dots, K-1.$$

Numerikus differenciálegyenlet-megoldás

A másik lehetőség π kiszámítására, hogy megoldjuk a mean-field differenciálegyenlet-rendszert numerikusan, majd behelyettesítünk egy nagy t értéket.

Következőnek azt nézzük meg, hogyan lehet differenciálegyenleteket numerikusan megoldani.

A legegyszerűbb esetben a

$$\frac{dx(t)}{dt} = F(x(t)) \quad x(0) = x_0$$

alakú egyenletet akarjuk megoldani, ahol $x(t)$ a (még ismeretlen) megoldás, $F(\cdot)$ az ún. vezérlő függvény, x_0 pedig a kezdeti feltétel. F és x_0 ismert.

Euler-módszer

Az Euler-módszer a következő: legyen $h > 0$ egy kicsi, de rögzített érték (pl. $h = 0.001$), ez lesz a lépésköz; ha már $x(t)$ -re van egy közelítésünk, akkor a $t + h$ pontban a megoldást a következő formulával közelítjük:

$$x(t + h) \approx x(t) + hF(x(t)).$$

Ezzel az $x(0) = x_0$ -ból indulva megkaphatjuk az $x(t)$ függvény egy közelítését a $h, 2h, \dots$ pontokban.

Mekkora a közelítés hibája? Az Euler-módszer egy lépésben h^2 nagyságrendű hibát vét; ha egy $[0, T]$ intervallumon szeretnénk a megoldást megkapni, akkor ehhez T/h lépésre van szükség, és a teljes hiba $[0, T]$ -n h nagyságrendű lesz.

Úgy is mondjuk, hogy az Euler-módszer elsőrendű módszer, mert a hibája h csökkentésével elsőrendben csökken.

Javított Euler-módszer

Lehet jobbat csinálni. A javított Euler-módszer (midpoint method) a következő közelítést használja:

$$x(t+h) \approx x(t) + hF\left(x(t) + \frac{h}{2}F(x(t))\right).$$

Ennek a hibája egy lépésben h^3 nagyságrendű, egy $[0, T]$ intervallum alatt pedig az össz hiba h^2 nagyságrendű; a javított Euler-módszer másodrendű módszer.

h csökkentésével mindkét módszer pontossága javul (de a javított Euler-módszeré gyorsabban), viszont az eljárás futásideje növekszik. Cserébe viszont a javított Euler-módszert bonyolultabb leködolni. A gyakorlatban az, hogy Euler-módszert vagy javított Euler-módszert érdemes használni, attól is függ, mennyire pontosan van szükségünk a megoldásra, és attól is, mennyire költséges az iterációt futtatni.

Vannak további módszerek is numerikus differenciálegyenlet-megoldásra, az általános keret Runge-Kutta módszerek néven fut.

Egyenlet-rendszer

Mi a helyzet, ha nem csak egy differenciálegyenletet szeretnénk megoldani, hanem egy differenciálegyenlet-rendszert?

$$\frac{dx_i(t)}{dt} = F_i(x(t)) \quad i = 1, \dots, K$$

Az Euler-módszer és a javított Euler-módszer is gond nélkül alkalmazható: az Euler-módszerre

$$x_i(t+h) \approx x_i(t) + hF_i(x(t)) \quad i = 1, \dots, K,$$

a javított Euler-módszerre

$$x_i(t+h) \approx x_i(t) + hF_i\left(x_i(t) + \frac{h}{2}F_i(x(t))\right) \quad i = 1, \dots, K.$$

Nem folytonos vezérlő függvény

Mi a helyzet, ha az F vezérlő függvény nem folytonos?

A szakadási ponton való áthaladást kezelniük kell. Ha az F függvény szakad egy y pontban, és

$$x(t) < y < x(t) + hF(x(t)),$$

akkor egy h nagyságú lépéssel már túlhaladnánk a szakadási ponton. Ehelyett legyen

$$h' = \frac{y - x(t)}{F(x(t))}.$$

Nem folytonos vezérlő függvény

Ha a t időpontból csak h' nagyságú lépést teszünk előre, akkor ez az $x(t)$ megoldást az F szakadási pontjába viszi, azaz

$$x(t + h') \approx y,$$

és onnantól előre felé már haladhatunk az F szakadási pont utáni értéke szerint; ha F -nek az y pontban a baloldali határértéke $F_-(y)$, a jobboldali $F_+(y)$, és felfelé léptük át a szakadási pontot F -ben, akkor

$$x((t + h') + h) \approx x(t + h') + hF_+(y),$$

míg ha lefelé léptük át a szakadási pontot, akkor $F_-(y)$ -t kell venni a jobb oldalon.

Vagy ha szeretnénk továbbra is az nh alakú pontokban közelíteni, akkor a h' nagyságú lépés után tehetünk még egy $h - h'$ nagyságú lépést. Azután haladhatunk tovább előre ismét h nagyságú lépésekkel.

Nem folytonos vezérlő függvény

A szakadási pont átlépését úgymond az Euler-módszer szerint kezeltük, ennek megfelelően egy átlépésnél h^2 nagyságrendű a közelítés hibája.

Egy $[0, T]$ intervallumon általában csak egy (vagy néhány, de fix számú) szakadási pont van, ezért a fenti átlépési módszert használhatjuk még akkor is, ha egyébként a folytonos részekben a javított Euler-módszert használjuk, mert az össz hiba így is h^2 nagyságrendű marad.