

## EGY KÉSLELTETETT DIFFERENCIÁLEGYENLET VIZSGÁLATA MEGBÍZHATÓ SZÁMÍTÓGÉPES ELJÁRÁSSAL

BÁNHELYI BALÁZS

A szakirodalomban nem volt ismert megbízható numerikus eljárás késleltetett differenciálegyenletek megoldására. Egy megbízható algoritmusnak nem csak a számításokban kell garantálnak lennie, de a formulákat is korrektül kell használnia. Jelen cikkben egy – sok szempontból is egyszerű – differenciálegyenleten keresztül próbáljuk illusztrálni az eljárás részleteit. A módszerünk alapja a kezdeti érték problémákra sokszor alkalmazott Taylor-sor. Ezzel az eljárással a függvény trajektóriája meghatározására egy megbízható befoglalást kapunk. A befoglaló intervallum ismeretében egy könnyen eldönthető feladat eredményét adjuk meg.

### 1. Bevezetés

Késleltetett differenciálegyenletekre nem ismert matematikai bizonyításokban is használható numerikus eljárás. Több gyakorlati probléma van, amelynek modellje visszavezethető ilyen késleltetett egyenletekre. A késleltetés természetes, ha arra gondolunk, hogy például az információ terjedése időbe telik, és így minden olyan döntés, válasz, vagy reakció – mely ezen információn alapul – késést indukál. Gondoljunk csak a biológiában a populációelméletre, orvostudományban a fertőzések terjedésére, vagy a közgazdaságtanban a részvények árfolyamának változására. Ezen témakörök tanulmányozása során felmerülhet bennünk az a kérdés, hogy a vizsgált paraméter által okozott ingadozó viselkedés állandósult, vagy csak a rövid vizsgálati idő miatt látjuk annak.

A jelen cikkben egy korábban felvetett [14] egyszerűnek tűnő problémát fogunk megvizsgálni. A feladatunk azt eldönteni, hogy egy, a differenciálegyenletek jobb oldalán szereplő paraméter függvényében hogyan viselkedik a differenciálegyenlet megoldása, trajektóriája. Hasonló jobboldalú, de szimmetrikus esetekre már léteznek eredmények [11]. Szimmetrikus abban az értelemben, hogy a differenciálegyenlet jobb oldalán szereplő – a megoldás függvényében megadott – függvény középpontosan szimmetrikus. A nonszimmetrikus esetekkel kapcsolatban vannak még bizonyítandó feltételezések.

E cikkben egy számítógépes vizsgálatot fogunk bemutatni. A programról megmutatjuk, hogy az matematikailag korrekt számításokat végez, és így egy számítógéppel segített matematikai bizonyítást fogunk adni a feltett kérdésre. Természetesen a teljes korrektség eléréséhez a formulákat is a megfelelő módon kell használni. Az eredményekből látni fogjuk, hogy a módszer működik, de az állítás teljes értékű bizonyításához a jelenlegi számítási kapacitások nem elegendők.

## 2. A vizsgált késleltetett differenciálegyenlet

Tekintsük az alábbi késleltetett differenciálegyenletet:

$$z'(t) = -\alpha z(t-1)(1+z(t)),$$

ahol  $\alpha \in \mathbb{R}^+$  paraméter. A megfelelő átalakításokkal csak késleltetést tartalmazó alakra hozható. Ha  $z(t) \geq -1$  (számunkra csak ez az eset érdekes), akkor vegyük a  $z(t) = e^{y(t)} - 1$  helyettesítést. Ekkor  $z'(t) = e^{y(t)}y'(t)$  és  $z(t-1) = e^{y(t-1)} - 1$ . Így az alábbi egyenletet kapjuk:

$$e^{y(t)}y'(t) = -\alpha \left( e^{y(t-1)} - 1 \right) \left( 1 + e^{y(t)} - 1 \right),$$

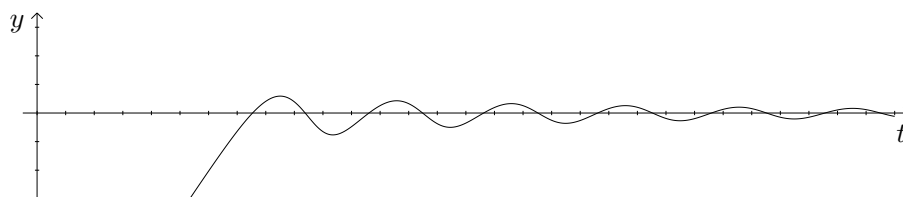
melyből egyszerűsítés után:

$$y'(t) = -\alpha \left( e^{y(t-1)} - 1 \right). \quad (1)$$

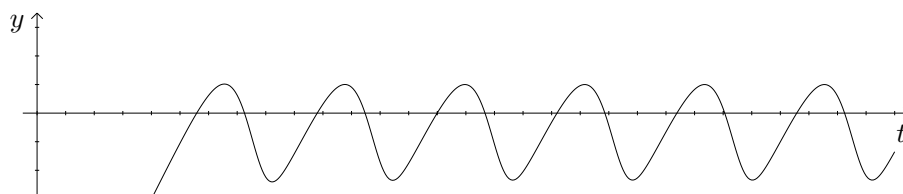
Az indulófüggvény legyen  $\phi(s)$ , ahol  $s \in [-1, 0]$ . Vizsgálatainkat a  $\phi(s) \equiv -1$  indulófüggvényre korlátozzuk.

Az  $\alpha \leq 1,5$  esetben ismert, hogy a trajektória oszcillálva konvergál a nullához (lásd az 1(a). ábrát). A  $\pi/2$ -ben bifurkáció lép fel, és megjelennek a periodikus pályák. Az  $\alpha > \pi/2$  esetén azonban már ezen periodikus pályák valamelyikéhez tart a megoldás. Ezért a továbbiakban csak az  $\alpha \in [1, 5, \pi/2]$  eseteket vizsgáljuk. Az eddigi numerikus eredmények alapján az sejtethető, hogy itt is hasonlóan viselkedik mint az  $\alpha \leq 1,5$  esetben.

A nullához való konvergencia vizsgálata numerikus módszerekkel nehézkes, így egy egyszerűsített problémát tanulmányozunk. A feladatunk az, hogy a késleltetett differenciálegyenlet megoldásairól eldöntsük, hogy létezik-e olyan  $a \in \mathbb{R}^+$  szám, hogy az  $[a, a+1]$  intervallumon a megoldás abszolút értéke kisebb, mint egy adott konstans. Ez az érték a mi esetünkben legyen  $0,075$ . A probléma így egyszerűbbé vált, mivel a nullához való konvergenciából következik a fenti tulajdonság. Ez a kettő nem ekvivalens állítás, mivel a fordított irány nem szükségszerűen igaz. A trajektória tarthat egy olyan periodikus megoldáshoz, melynek szélsőértékei a  $\pm 0,075$  között vannak, és nem nullák (lásd az 1(b). ábrát az  $\alpha = 2,0$  esetre).



(a) Az  $\alpha = 1,5$  eset.



(b) Az  $\alpha = 2,0$  eset.

**1. ábra.** Közelítő ábrák a trajektóriára.

### 3. A differenciálegyenlet mélyebb vizsgálata

Több módszer is ismert a differenciálegyenletek megoldásainak közelítésére. A hagyományos differenciálegyenletekre kifejlesztett matematikai bizonyításokban is használható [7] módszerek egy jó része a Taylor-soron alapul [4, 13]. Így mi is ezen az elven működő eljárást alkalmazunk a jelenlegi késleltetett differenciálegyenletre.

Lagrange-féle maradéktaggal ellátott Taylor-polinom:

$$y(x) = \sum_{k=0}^{n-1} \frac{(x-x_0)^k y^{(k)}(x_0)}{k!} + r_n, \tag{2}$$

ahol

$$r_n = \frac{(x-x_0)^n}{n!} y^{(n)}(x^*), \tag{3}$$

valamely  $x^* \in [x_0, x]$ -re ( $x_0 \leq x$ ).

Ha elhagyjuk az  $r_n$  maradéktagot a képletből, akkor egy közelítést kapunk  $y(x)$ -re. A kapott formulát megvizsgálva látható, hogy a magasabbrendű deriváltak alkalmazása esetén jobban közelíthető a függvény. A következő állítás segítségével határozhatjuk meg a magasabbrendű deriváltakat.

**3.1. ÁLLÍTÁS.** Az (1)-es késleltetett differenciálegyenlet magasabbrendű deriváltjai ( $k \geq 2$ ) felírhatók a következő formulával:

$$y^{(k)}(t) = -\alpha y^{(k-1)}(t-1) + \sum_{i=1}^{k-1} \binom{k-2}{i-1} y^{(i)}(t-1) y^{(k-i)}(t). \tag{4}$$

*Bizonyítás.* A bizonyítást  $k$  szerinti teljes indukcióval végezzük. Tekintsük először a  $k = 2$  esetet:

$$\begin{aligned}
y^{(2)}(t) &= \left( -\alpha \left( e^{y(t-1)} - 1 \right) \right)' = -\alpha e^{y(t-1)} y'(t-1) = \\
&= y'(t-1) \left( -\alpha \left( e^{y(t-1)} - 1 \right) \right) - \alpha y'(t-1) = \\
&= y'(t-1) y'(t) - \alpha y'(t-1) = \\
&= -\alpha y^{(1)}(t-1) + \binom{0}{0} y^{(1)}(t-1) y^{(1)}(t) = \\
&= -\alpha y^{(2-1)}(t-1) + \binom{2-2}{1-1} y^{(1)}(t-1) y^{(2-1)}(t) = \\
&= -\alpha y^{(k-1)}(t-1) + \sum_{i=1}^{k-1} \binom{k-2}{i-1} y^{(i)}(t-1) y^{(k-i)}(t).
\end{aligned}$$

Tegyük fel, hogy valamely  $k \geq 2$ -re igaz az állítás. Nézzük a  $k+1$  esetet.

$$\begin{aligned}
y^{(k+1)}(t) &= \left( y^{(k)}(t) \right)' = \\
&= \left( -\alpha y^{(k-1)}(t-1) + \sum_{i=1}^{k-1} \binom{k-2}{i-1} y^{(i)}(t-1) y^{(k-i)}(t) \right)' = \\
&= -\alpha y^{(k-1+1)}(t-1) + \\
&+ \sum_{i=1}^{k-1} \binom{k-2}{i-1} \left( y^{(i+1)}(t-1) y^{(k-i)}(t) + y^{(i)}(t-1) y^{(k-i+1)}(t) \right)
\end{aligned}$$

Mivel az első tag már azonos, így vizsgálatainkat folytassuk a második taggal.

$$\begin{aligned}
&\sum_{i=1}^{k-1} \binom{k-2}{i-1} \left( y^{(i+1)}(t-1) y^{(k-i)}(t) + y^{(i)}(t-1) y^{(k-i+1)}(t) \right) = \\
&= \sum_{i=1}^{k-1} \binom{k-2}{i-1} \left( y^{(i+1)}(t-1) y^{(k-i)}(t) \right) + \\
&+ \sum_{i=1}^{k-1} \binom{k-2}{i-1} \left( y^{(i)}(t-1) y^{(k-i+1)}(t) \right) = \\
&= \binom{k-2}{k-1-1} \left( y^{(k-1+1)}(t-1) y^{(k-k+1)}(t) \right) + \\
&+ \sum_{i=1}^{k-2} \binom{k-2}{i-1} \left( y^{(i+1)}(t-1) y^{(k-i)}(t) \right) +
\end{aligned}$$

$$\begin{aligned}
 & + \binom{k-2}{1-1} \left( y^{(1)}(t-1)y^{(k-1+1)}(t) \right) + \\
 & + \sum_{i=2}^{k-1} \binom{k-2}{i-1} \left( y^{(i)}(t-1)y^{(k-i+1)}(t) \right) = \\
 = & \binom{k+1-2}{k+1-2} \left( y^{(k+1-1)}(t-1)y^{(k+1-k)}(t) \right) + \\
 & + \binom{k+1-2}{0} \left( y^{(1)}(t-1)y^{(k+1-1)}(t) \right) + \\
 & + \sum_{i=2}^{k-1} \binom{k-2}{i-1-1} \left( y^{(i)}(t-1)y^{(k-i+1)}(t) \right) + \\
 & + \sum_{i=2}^{k-1} \binom{k-2}{i-1} \left( y^{(i)}(t-1)y^{(k-i+1)}(t) \right) = \\
 = & \binom{k+1-2}{k+1-2} \left( y^{(k+1-1)}(t-1)y^{(k+1-k)}(t) \right) + \\
 & + \binom{k+1-2}{0} \left( y^{(1)}(t-1)y^{(k+1-1)}(t) \right) + \\
 & + \sum_{i=2}^{k-1} \binom{k+1-2}{i-1} \left( y^{(i)}(t-1)y^{(k-i+1)}(t) \right)
 \end{aligned}$$

Azaz

$$y^{(k+1)}(t) = -\alpha y^{(k+1-1)}(t-1) + \sum_{i=1}^{k+1-1} \binom{k+1-2}{i-1} \left( y^{(i)}(t-1)y^{(k+1-i)}(t) \right),$$

mellyel az állítást bizonyítottuk. □

Vegyük észre, hogy az (1)-es egyenletben csak az  $y(t-1)$  érték szerepel, így a  $t$  időpillanatban az első derivált kiszámítható a  $(t-1)$  pontbeli függvényérték ismeretében. A magasabbrendű deriváltak kiszámításához elegendő a  $t$  és a  $(t-1)$  időpillanatban az alacsonyabbrendű deriváltak értéke. Tehát a  $t$  időpillanatban az összes deriváltat ki tudjuk számolni az alacsonyabbrendű deriváltaktól haladva a magasabbrendűek felé.

Az alábbi tétel szerint a megoldás létezéséhez elégséges feltétel, hogy a függvény folytonos legyen.

**3.1. TÉTEL.** *Tekintsük az  $y'(t) = f(y(t-1))$  ( $t \geq 0$ ) késleltetett differenciálegyenletet, a  $\phi(t)$  ( $t \in [-1, 0]$ ) indulófüggvénnyel. Folytonos jobboldalú és folytonos indulófüggvénnyel adott explicit késleltetett differenciálegyenletnek létezik megoldása.*

*Bizonyítás.* A tétel állítása hasonló a Cauchy-Peano-féle egzisztenciátételhez, de mivel mi most egy tisztán késleltetett differenciálegyenletről állítjuk ugyanazt, ezért a bizonyítás is egyszerűbb.

Elegendő a  $[0, 1]$  időintervallumon vizsgálni a megoldást. Ha ezen intervallumon létezik folytonos megoldása, akkor a következő 1 hosszú intervallumra ugyanezt a bizonyítási eljárást alkalmazhatjuk.

Vegyük észre, hogy a  $t \in [0, 1]$  időpontban a függvény értéke az alábbi formulával számolható:

$$y(t) = y(0) + \int_0^t y'(x) dx = y(0) + \int_{-1}^{t-1} -\alpha(e^{y(x)} - 1) dx.$$

Ha tehát  $y'$  integrálható, akkor az  $y$  egy folytonos függvény lesz a  $[0, 1]$  intervallumon. Ehhez az kell, hogy  $y'(t)$  véges sok pont kivételével ne legyen folytonos. Ez pedig következik abból, hogy  $y(t)$  folytonos a  $[-1, 0]$  intervallumon.  $\square$

A tételben szereplő  $f$  függvény folytonos, így mindig létezik egy megoldás, és a Taylor-polinom közelítésként alkalmazható rá. Ezek után azt vizsgáljuk meg, hogy hol lehetnek szakadási pontjai az (1)-es differenciálegyenletnek.

**3.2. ÁLLÍTÁS.** Az (1)-es differenciálegyenlet szakadási pontjaira az alábbiak igazak:

- a differenciálegyenletnek és magasabbrendű deriváltjainak csak az egész-értékű pontokban lehet szakadási pontja,
- az  $i$ . deriváltnak az  $(i - 1)$ -nél nagyobb egész pontokban már nem lehet szakadási pontja ( $i \geq 0$ ).

*Bizonyítás.* Az állítás bizonyításhoz a 3.1. Állításban szereplő képletet, valamint az (1)-es formulát kell megvizsgálni. Vegyük észre, hogy a vizsgált indulófüggvény is, és a deriváltjai is a  $[-1, 0)$  intervallumon folytonosak. Legyen a vizsgált pont  $t \geq 0$ . A folytonos függvények megadott kombinációja folytonos függvény. A deriváltakra vonatkozó képletekben csak a  $t$  és  $(t - 1)$  pontok szerepelnek. Valamely alacsonyabbrendű deriváltnak ezen pontok valamelyikében szakadási pontjának kell lennie ahhoz, hogy szakadási pontja legyen a tekintett derivált függvénynek. Így ha van szakadási pont, akkor annak periódusa 1. A tekintett indulófüggvény első deriváltjának csak a 0-ban van szakadási pontja, így az első állítás igaz.

A második állítás igazolásához is elegendő a képletek alapján a folytonosságot vizsgálni. Az  $i = 0$  esetben a függvényről állítjuk, hogy folytonos, melyet már láttunk az előző tételben. A magasabbrendű deriváltak esetében a képleteket megvizsgálva látható, hogy az  $i$ . deriváltnak, akkor lesz szakadási pontja, ha a  $0 \dots (i - 1)$  deriváltak közül legalább az egyiknek van szakadási pontja a  $t$  vagy a  $(t - 1)$  pontok valamelyikében. Az indulófüggvény folytonos, így az első derivált is folytonos lesz 0 után, a második derivált pedig folytonos lesz 1 után és így tovább. Tehát a második állítás is igaz.  $\square$

Mivel az első derivált folytonos a  $t > 0$  időpillanatokban, így korlátos is. Vegyük észre, hogy az első derivált a  $t = 0$  időpillanatban is korlátos. E két tulajdonságból következik, hogy az első derivált korlátos minden  $[-1, t']$  időintervallumon, ahol  $t' \geq -1$ . Ebből a tulajdonságból következik a differenciálegyenlet megoldásának egyértelmősége is, de jelen cikkben ezt nem használjuk ki.

#### 4. Megbízható numerikus módszerek

A számítógépeken történő számábrázolások elterjedt, hatékony formája a lebegőpontos számábrázolás. Ezzel csak bizonyos pontossággal tudunk számolni. Előfordulhat, hogy a számítások során az eredményt nem tudjuk pontosan ábrázolni számítógép segítségével. Egyik lehetőség, hogy az eredmény egy intervallumos befoglalását adjuk meg. Ekkor valós számok helyett intervallumokat használunk, melyekre definiálni kell a műveleteket. Példaként definiáljuk a négy alapműveletet.

##### 4.1. Definíció. (intervallumaritmetika)

$$A \circ B = \{a \circ b \mid a \in A \text{ és } b \in B\},$$

$A, B \in \mathbb{I}$  ( $\mathbb{I}$  az  $(i, j)$  párok halmaza, ahol  $i, j \in \mathbb{R}$ , és  $i < j$ ).

Ezt a következő konstruktív módon lehet megvalósítani:

$$\begin{aligned} [a, b] + [c, d] &= [a + c, b + d], \\ [a, b] - [c, d] &= [a - d, b - c], \\ [a, b] \cdot [c, d] &= [\min(ac, ad, bc, bd), \max(ac, ad, bc, bd)], \\ [a, b] / [c, d] &= [a, b] \cdot [1/d, 1/c], \text{ ha } 0 \notin [c, d]. \end{aligned}$$

Az intervallumok számítógépes ábrázolása során újabb probléma áll elő, ha az eredményintervallum határai nem ábrázolhatók számítógépen. Ezért kifele kerekítjük a határpontokat, azaz vesszük a legközelebbi számítógépen ábrázolható számot. Ekkor az eredmény egy megbízható befoglalását kapjuk, azaz az eredmény intervallum tartalmazza a pontos eredményt.

Ilyen megbízható műveletek sorozatával megadható a függvényérték egy garantált befoglalása. Így az az elvárásunk, hogy egy többváltozós, intervallumos függvény eredményintervalluma minden többdimenziós intervallum esetén tartalmazza a valós függvénynek az adott helyen számított értékkészletét.

4.2. Definíció.  $F(\mathbb{I}^n \rightarrow \mathbb{I})$  befoglaló függvénye  $f(\mathbb{R}^n \rightarrow \mathbb{R})$ -nek az  $\mathbf{X}$  többdimenziós intervallumon, ha minden  $\mathbf{Y} \subseteq \mathbf{X}$ -re

$$f(\mathbf{Y}) = \{z : z \in \mathbb{R} \mid z = f(\mathbf{y}) \text{ és } \mathbf{y} \in \mathbf{Y}\} \subseteq F(\mathbf{Y})$$

teljesül.

Megjegyzendő, hogy a megbízható műveletek sorozatával kiértékelt összetett függvények eredményintervalluma sokszor bővebb lesz, mint a valódi értékkészlet, úgynevezett túlbecslést eredményező intervallum lesz. A másik probléma, hogy nem minden függvény értékelhető ki véges számú aritmetikai művelet elvégzésével. Gondoljunk csak a  $\sqrt{x}$ -re, a  $\sin x$ -re, és a jelenlegi problémánkban használt  $e^x$ -re. Ezen függvények kiértékelésére az egyik lehetőség a jelen cikkben is alkalmazott módszer lehet. Így például az  $e^x$ -re az alábbi módszert alkalmazzák:

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!} \in \sum_{n=0}^k \frac{x^n}{n!} + [-2, 2] \frac{x^{k+1}}{(k+1)!},$$

ha  $0 \leq x \leq 1$ . Részletesebben olvashatunk az intervallumaritmetikáról az [1, 12] dolgozatokban.

Az intervallumos műveleteket támogató intervallumos könyvtárak léteznek C és C++ környezetben [8, 10], melyeket több esetben is használtunk matematikai bizonyításokban [2, 3, 6]. A jelen problémát mind a C-XSC, mind a Profil/Bias könyvtár használatával megvalósítottuk. Mindkét könyvtár az „Interval” típus végpontjait C-beli double típussal ábrázolja. A végpontok pontossága befolyásolja az intervallumaritmetika pontosságát. Esetünkben ez az alap típus nem volt elegendően pontos. Mindkét könyvtárban megtalálható azonban egy nagyobb precizitást szolgáló intervallum típus. A C-XSC a végpontokat több double típussal írja le [8], és ezek száma beállítható. Míg a Profil/Bias esetén a kívánt pontosságot lehet megadni [9].

## 5. Az intervallumos befoglalás használata

Az intervallumaritmetika segítségével megbízható számítások végezhetőek, így alkalmas bizonyos problémák matematikai erejű bizonyítására. Jelen esetben a bizonyításhoz a Lagrange-féle maradéktaggal ellátott Taylor-polinom intervallumos megvalósítása jól használható. A matematikai bizonyításhoz a formulákat megbízható alakban, intervallumos befoglaló függvényekkel kell használni. A módszerünk alapja, hogy az  $r_n$ -re megpróbálunk egy befoglaló intervallumot adni. Ennek segítségével már be tudjuk foglalni a  $y(x)$  megoldást.

5.1. ÁLLÍTÁS. *Az  $x$  időpontban ( $x \geq x_0$ ) a függvényérték befoglalására, azaz  $y(x) \in Y(x)$ , a Lagrange-féle maradéktaggal ellátott Taylor-polinom az alábbi formában alkalmazható:*

$$Y(x) = \sum_{k=0}^{n-1} Y^{(k)}(x_0) \frac{(x-x_0)^k}{k!} + R_n,$$

ahol

$$R_n = Y^{(n)}([x_0, x]) \frac{(x-x_0)^n}{n!}.$$



*Bizonyítás.* Egy konkrét pontban a függvényérték befoglalásához az  $r_n$  értéket kell befoglalni, mivel a (2)-es képlet első tagja egyszerűen meghatározható. Természetesen e tag kiértékelését is megbízhatóan kell elvégezni, hogy garantált befoglalást kapjunk rá. A (3)-as formula szerint létezik olyan  $x^* \in [x_0, x]$ , melyre  $r_n$  pontos. Ha az  $r_n$  befoglalását a (3)-as formulával határozzuk meg az  $[x_0, x]$  intervallumon, akkor az  $Y(x)$  befoglalás korrekt lesz, azaz  $y(x) \in Y(x)$ .  $\square$

Az 5.1. Állítás alkalmazásához szükség van az  $Y^{(n)}([x_0, x])$  értékre. Ez egy magasabbrendű derivált, melyre láttuk, hogy meghatározásához szükség van  $y(t-1)$ -re, azaz szükségünk van az  $Y([x_0-1, x-1])$  befoglalásos intervallumára. Az előző módszer megfelelő módosításokkal működik az  $[x_0, x]$  intervallumon is.

5.2. ÁLLÍTÁS. Az  $[x_0, x]$  időintervallumon a függvényérték befoglalására, azaz  $y([x_0, x]) \subseteq Y([x_0, x])$ , a Lagrange-féle maradéktaggal ellátott Taylor-polinom az alábbi formában alkalmazható:

$$Y([x_0, x]) = \sum_{k=0}^{n-1} Y^{(k)}(x_0) \frac{([0, x-x_0])^k}{k!} + R_n,$$

ahol

$$R_n = Y^{(n)}([x_0, x]) \frac{([0, x-x_0])^n}{n!}.$$

*Bizonyítás.* Tekintsünk egy tetszőleges  $x' \in [x_0, x]$  pontot. Ekkor azt kell belátni, hogy az  $Y([x_0, x])$  befoglalás tartalmazza az  $y(x')$  értéket. Az előbbieken láttuk, hogy az  $Y(x')$  intervallumnak eleme az  $y(x')$  érték. Vegyük észre, hogy az  $[x_0, x']$  intervallum részintervalluma az  $[x_0, x]$ -nek, illetve  $(x'-x_0)$  része a  $[0, x-x_0]$  intervallumnak. Ezen befoglalásokat felhasználva látható, hogy az  $Y([x_0, x])$  befoglaló intervalluma tartalmazza az  $Y(x')$  értéket, mellyel az állítást igazoltuk.  $\square$

5.1. TÉTEL. A fenti formulákkal megbízható befoglalás adható az (1)-es differenciálegyenlet megoldására.

*Bizonyítás.* Egy teljes indukcióhoz hasonló gondolatmenettel belátható a tétel. A  $[-1, 0]$  intervallum minden pontjában ismert a  $0, \dots, n$ -edik deriváltak egy-egy befoglalása. Tegyük fel, hogy minden  $-1 \leq t \leq x$ -re tudjuk a függvény és a deriváltak befoglalását. Tekintsünk egy tetszőleges  $x_0 \in [x, x+1]$  pontot. Ekkor az első deriváltak befoglalása kiszámítható az  $x_0$  pontban és az  $[x, x_0]$  intervallumon az (1)-es képlettel. Mivel a magasabbrendű deriváltak meghatározásához elegendő az első derivált ismerete az  $[x, x_0]$  intervallumon, valamint a magasabbrendű deriváltak az  $[x-1, x_0-1]$ -en, így a 3.1. Állításban szereplő formulával számítható a magasabbrendű deriváltak befoglalása. A deriváltak ismeretében az 5.1. és az 5.2. Állítások alkalmazásával számítható a függvény egy befoglalása az  $x$  pontban és az  $[x, x_0]$  intervallumon. Így  $x_0$ -ig ismert a  $0, \dots, n$ -edik deriváltak egy befoglalása.  $\square$

A magasabbrendű deriváltak képlettel számíthatóak, és minden  $t \geq -1$  pontban létezik megoldás. Így a bizonyításon alapuló eljárással egy adott indulófüggvényből bármely  $t \geq 0$  pontban számolható a megoldás egy befoglalása.

Visszatérve a kezdeti problémához, azt kell megvizsgáljunk, hogy van-e olyan 1 hosszú szakasza a megoldás befoglalásának, amely teljes egészében a  $[-0,075, 0,075]$  intervallumba esik. Ez a kérdés megbízhatóan eldönthető a befoglaló intervallumok végpontjainak vizsgálatával. Most nézzük meg, hogy milyen esetben nincs értelme folytatni a trajektória követését. Ehhez előbb a következő állítást fogjuk igazolni.

**5.3. ÁLLÍTÁS.** *A trajektória befoglalásának szélessége az  $x_0$  pontban ( $x_0 > x$ ) nem kisebb, mint az  $x$ -ben.*

*Bizonyítás.* Mivel  $x_0 > x$ , így az  $x_0$  pont eléréséhez minimum egy lépés szükséges az  $x$  pontból. Így elegendő, ha belátjuk, hogy egyetlen lépésben sem csökken a befoglalás szélessége. Vegyük észre, hogy ha egy intervallumhoz intervallumaritmetika segítségével hozzáadunk egy másikat, akkor az eredeti intervallumnál nem kapunk kisebb szélességűt. Ugyanakkora szélességűt abban az esetben kapunk, ha a hozzáadott intervallum szélessége 0. Most vizsgáljuk meg az 5.1. Állításban szereplő képletet. Az összegzést kibontva pontosan  $n$  darab összeadás szerepel a képletben. A  $k = 0$  esetben az  $Y(x)$ -et kapjuk, melyhez már csak további intervallumokat adunk hozzá. Ezzel igazoltuk, hogy nem csökken a befoglalás szélessége.  $\square$

Mivel a megoldásnak abszolút értékben kisebbnek kell lennie  $0,075$ -nél, így a megfelelő sáv szélessége  $2 \cdot 0,075$ . Az állításból következik, hogy ha az  $x$  pontban a függvényérték befoglaló intervallumának szélessége legalább  $2 \cdot 0,075$ , akkor az  $x_0 > x$  pontokban nem lesz kisebb. Így az  $x$ -nél nagyobb pontokban a feladatnak megfelelő szakaszt már nem találhatunk. Azaz a trajektória további követése nem vezethet sikerre.

## 6. Az ellenőrző eljárás

A függvény és derivált értékek befoglalását az  $x_i$  időpillanatban is és a  $t_i = [x_{i-1}, x_i]$  időintervallumon is tároljuk. A trajektória követését az  $x_i - x_{i-1} = c$  fix lépésközzel valósítjuk meg minden  $i$ -re, ahol  $c$  egy konstans. A számítógép pontosságára tekintettel, ezt a  $c$  konstanszt a számítógépen ábrázolható számok közül választjuk ki. A befoglalások számításához az  $x_i$ -ben, valamint a  $t_i$ -ben az  $(x_i - 1)$  és a  $(t_i - 1)$  időbeli értékekre van szükség. Ezért érdemes a  $c$  konstanszt úgy megválasztani, hogy ha az  $x_i$ -ben van végpont, akkor az  $(x_i + 1)$ -ben is legyen. Így a befoglalásokat nem kell újra számolni több befoglalásból. A  $c$  konstans minden esetben  $2^{-n}$  alakú, ahol  $n \in \mathbb{N}^+$ . A kezdeti 1 hosszú szakaszon a függvény értékére és deriváltjaira könnyen adható befoglalás minden  $x_i$  és  $t_i$  időkből.

Két listában tároljuk az  $x_i$  és a  $t_i$  időkből a függvény és a derivált értékek befoglalását. A két listát a későbbiekben  $\mathcal{X}$ -szel és  $\mathcal{T}$ -vel jelöljük. Ezen függvény-leírásból mindkét listában pontosan  $n$  darab lesz. A  $\mathcal{T}$  lista összes eleme pontosan 1 hosszú szakaszt tartalmaz, mely minden egyes pontjában befoglalja a függvényt. Ez elegendő a feltett kérdés megválaszolásához.

Az eljárás a listákat bővíteni fogja a következő  $x_i$  és  $t_i$  időkből a függvény és a derivált értékek befoglalásával. Az 5.1. Tételből következik, hogy a befoglalások kiszámításához elegendő a  $\mathcal{T}$  lista első eleme, és az  $\mathcal{X}$  lista első és utolsó eleme. Az új befoglalásokat betesszük a listák végére, és az első elemet töröljük a listákból. A  $\mathcal{T}$  lista megint egy teljes, 1 hosszú időintervallumon írja le a függvényt.

Ezekkel a megkötésekkel elértük, hogy minden lépés előtt az alábbi állítások igazak:

- (1) A  $\mathcal{T}$  lista első eleme tartalmazza az  $N$ . lépésben az  $y^{(i)}((N - 2^n)/2^n, (N - 2^n + 1)/2^n)$  ( $i = 0, \dots, K$ ) deriváltak befoglalásait.
- (2) Az  $\mathcal{X}$  lista első eleme tartalmazza az  $N$ . lépésben az  $y^{(i)}((N - 2^n + 1)/2^n)$  ( $i = 0, \dots, K$ ) deriváltak befoglalásait.
- (3) Az  $\mathcal{X}$  lista utolsó eleme tartalmazza az  $N$ . lépésben az  $y^{(i)}(N/2^n)$  ( $i = 0, \dots, K$ ) deriváltak befoglalásait.

A  $\mathcal{T}$  és az  $\mathcal{X}$  lista következő elemeinek a meghatározásához éppen ezekre az értékekre van szükségünk. Ellenőrzéskor elegendő az új

$$Y([(N - 2^n)/2^n, (N - 2^n + 1)/2^n])$$

befoglalást megvizsgálni, és ha arra igaz az állítás, akkor egy számlálót növelni, egyébként nullázni. Ha a számláló elérte az 1 hosszú szakaszhoz szükséges értéket, akkor megtaláltuk a szakaszt, amely igazolja a feltételezést. Az 5.3. Állításból ismert, hogy a befoglalás szélessége a lépések során nem csökken. Tudjuk, hogy ha az  $Y(N/2^n)$  befoglalás szélessége nagyobb, mint  $2 \cdot 0,075$ , akkor már nem fog beleférni a kívánt sávba. Ekkor az algoritmus sikertelen kereséssel áll meg.

*6.1. Algoritmus.* A trajektória követése.

- Input:*
- $K$ : a használt legmagasabbrendű derivált rendje,
  - $1/2^n$ : a lépés nagysága,
  - $k = 0,075$ : a feladat kitűzésében szereplő konstans,
  - $\phi(s) \equiv -11$ : az indulófüggvény.

- Output:*
- Az  $[a, a + 1]$  intervallum, melyen a függvényérték abszolút értéke kisebb mint  $k$ , vagy
  - nem talált megfelelő intervallumot.

**0. lépés:** Töltsük fel a listákat a  $[-1, 0]$  szakaszon levő befoglalásokkal, és legyen  $N = 0$ .

**1. lépés:** Vegyük ki a listák első elemét, és az  $\mathcal{X}$  lista utolsó elemét.

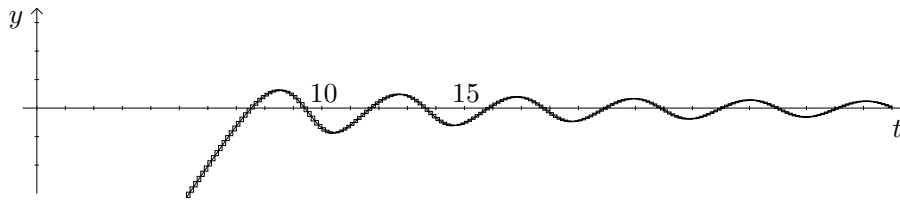
2. lépés: Számítsuk ki a (4)-es képlet használatával az  $[N/2^n, (N+1)/n]$  intervallumon az  $Y^{(i)}$  ( $i = 1 \dots K$ ) deriváltakat.
3. lépés: Határozzuk meg a (4)-es képlet felhasználásával az  $(N+1)$  pontban az  $Y^{(i)}$  ( $i = 1 \dots K$ ) deriváltakat.
4. lépés: Számítsuk ki az 5.1. Állítás segítségével az  $[N/n, (N+1)/n]$  intervallumon az  $Y$  értékeket.
5. lépés: Határozzuk meg az 5.2. Állítás segítségével az  $(N+1)/n$  pontban az  $Y$  értéket.
6. lépés: Frissítsük a két listát.
7. lépés: Legyen  $N$  új értéke  $N+1$ .
8. lépés: Ellenőrizzük az új  $Y([N/2^n, N/n])$  befoglalást. Ha megfelel az  $|x| \leq k$  feltételnek és elértük az 1 hosszú szakaszt, akkor az  $[N/n - 1, N/n]$  intervallum a megoldás, és STOP.
9. lépés: Ha az  $Y((N+1)/n)$  pontban a befoglalás nagyobb mint  $2 \cdot k$ , akkor nem talált megfelelő intervallumot és STOP.
10. lépés: Folytassuk az 1. lépéssel.

Az algoritmus az 5.1. Tétel bizonyítása alapján készült, és minden számítása garantált. Az eljárás sikeres futása esetén matematikai bizonyossággal állíthatjuk, hogy létezik egy 1 hosszú szakasz a feladat megoldásaként. Egyébként nem állíthatjuk azt, hogy nem létezik ilyen szakasz, azaz a jelenleg használt számítási paraméterekkel a befoglalások növekedése miatt nem volt sikeres a kísérlet. Ebben az esetben nem zárható ki a feladatra egy ilyen szakasz létezése. Elméletileg előfordulhat, hogy az intervallum szélessége soha nem lesz a korlátnál nagyobb, és nem talál 1 hosszú szakaszt sem. Ekkor az eljárás egy végtelen ciklusba kerülne, de ez a számítógép véges pontosságát tekintve nem fordulhat elő.

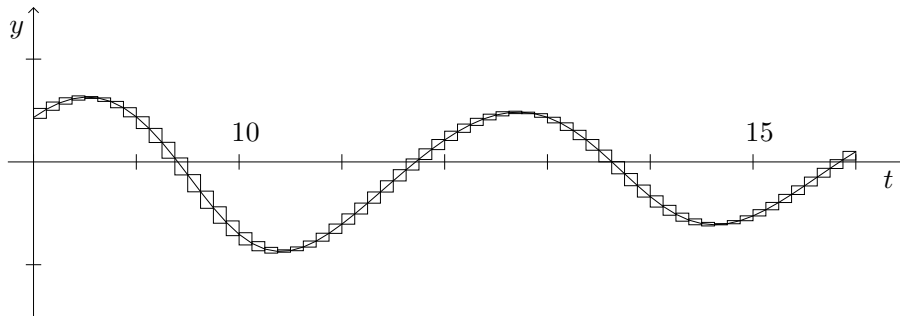
## 7. Eredmények

Első lépésben az  $\alpha = 1,5$  esetet vizsgáltuk, melyre ismert, hogy a megoldás nullához tart. Ekkor találnunk kell egy megfelelő 1 hosszú szakaszt. A 25. deriváltig számoltuk a befoglalásokat, és a lépésmagyságot  $2^{-3} = 1/8$ -nak választottuk. A dupla pontosságú adattípussal ábrázolt végpontú intervallum típus nem volt elegendő. Ehhez, az általunk vizsgált legegyszerűbb feladathoz is már szükség volt a nagyobb pontosságot használó intervallumaritmetikára. Az első megfelelő szakasz az  $[56, 57]$  volt, amelyre  $|Y([56, 57])| \leq 0,075$ . Ennek megtalálásához 25 másodperc CPU időre volt szükség egy egyszerű asztali számítógépen (Pentium IV, 2,2 GHz-es processzor).

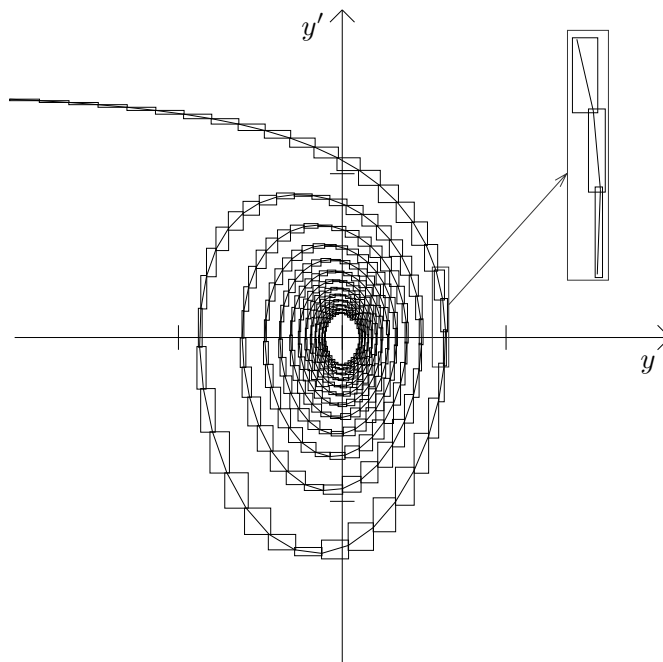
Az 2. ábrán a trajektória befoglalását láthatjuk az idő függvényében. Mivel a lépés nagysága  $2^{-3}$ , így egy 1 hosszú szakasz felett pontosan 8 darab befoglalás található. A szomszédos befoglaló téglalapok függőleges oldalai pontosan egy egyenesre illeszkednek, és ezek az oldalak nem csatlakoznak a végpontjaikban, hanem átfedik egymást. Ez az átfedés tartalmazza az adott időben a függvényérték



(a) A teljes trajektória egy részlete.



(b) Az (a) ábrán látható befoglalás egy kinagyított részlete.



(c) A teljes trajektória az  $y$  és  $y'$  térben.

**2. ábra.** A trajektória közelítése az  $\alpha = 1,5$  esetben és annak garantált befoglalása.

befoglalását. A 2(c). ábrán ugyanaz a trajektória látható. A téglalapok ebben az esetben az  $y$  függvényérték és az  $y'$  első derivált befoglalását ábrázolják.

Az 1. táblázatban látható, hogy azonos deriváltak használata mellett, ha a lépéshossz felére csökken, akkor a szükséges idő kb. kétszeresére nő. Ennek a magyarázata egyszerű, mert pontosan kétszerannyi lépést kell végrehajtani, mint a kétszeres lépéshossz esetén. A másik észrevehető tulajdonság, hogy az alkalmazott deriváltak maximális számának csökkentésével csökken a szükséges CPU idő is. Ennek magyarázata a magasabbrendű deriváltak számításához használt egyre összetettebb képlet. De látható az is, hogy mindkettő túlzott csökkentésével a bizonyítás nem sikerül.

A deriváltak száma	Lépéshossz								
	$2^{-1}$	$2^{-2}$	$2^{-3}$	$2^{-4}$	$2^{-5}$	$2^{-6}$	$2^{-7}$	$2^{-8}$	$2^{-9}$
5	X	X	X	X	X	X	X	X	X
10	X	X	X	X	X	X	105	213	431
15	X	X	X	X	45	93	187	373	771
20	X	X	X	35	73	150	291	585	1157
25	X	X	25	53	107	212	424	851	1695
30	X	X	36	72	146	292	584	1169	2314
35	X	23	48	96	191	383	765	1530	3055
40	X	30	61	123	245	491	980	1956	3903
45	X	38	77	153	306	611	1220	2437	4862

**1. táblázat.** Szükséges CPU idő (másodpercben) az  $\alpha = 1,5$  eset vizsgálatakor. A táblázat sorai az összes használt derivált maximális rendjét, míg az oszlopok a lépés nagyságát jelentik. Az X-szel jelölt esetekben a bizonyítás nem sikerült.

Megvizsgáltuk, hogy milyen pontosság érhető el a számítások során használt intervallumok végpontjainak pontosságával, melyek a 2. táblázatban láthatók. Az  $\alpha = 1,5$  esetben különböző paraméterekkel követtük a trajektóriát a  $[0, 50]$  időintervallumon. Megnéztük, hogy ez mennyi CPU időt igényelt, és mennyire pontos a függvényérték befoglalása a  $t = 50$  időpillanatban. Látható, hogy a kívánt számítási pontosság növelésével nőtt a szükséges CPU idő. Azonban az elérhető pontosságot nem tudjuk kihasználni, ha nem használunk magasabbrendű deriváltakat. A számítási pontosságot nem változtatva, csak magasabbrendű derivált használatával, nem tudunk nagyobb pontosságot elérni. Így minden számítási pontossághoz tartozik egy optimális deriváltrend, és minden használni kívánt deriválthoz létezik egy optimális számítási pontosság. Azaz a számítási pontosságot

növelve magasabbrendű deriváltakat is használhatunk, de ekkor a CPU idő is nő. Míg az 1. táblázatban láthattuk, hogy minden lépéshozhoz használni kell egy minimális nagyságú rendet. Más szóval, a magasabbrendű deriváltak használatával csökkenthetjük a lépéshossz nagyságát, mellyel a CPU idő is csökken. Összegezve láthatjuk, hogy a szükséges CPU idő a lépés nagyságtól és a számítási pontosságtól függ, melyek fordított arányban állnak egymással.

A deriváltak száma	Az intervallumok k dupla pontos számítással			
	k = 1	k = 2	k = 4	k = 8
15	6 sec $2,78 \cdot 10^0$	9 sec $2,01 \cdot 10^{-1}$	12 sec $2,01 \cdot 10^{-1}$	21 sec $2,01 \cdot 10^{-1}$
20	11 sec $2,58 \cdot 10^0$	15 sec $7,70 \cdot 10^{-5}$	19 sec $7,70 \cdot 10^{-5}$	32 sec $7,70 \cdot 10^{-5}$
25	17 sec $2,58 \cdot 10^0$	23 sec $2,86 \cdot 10^{-9}$	28 sec $2,86 \cdot 10^{-9}$	46 sec $2,86 \cdot 10^{-9}$
30	24 sec $2,58 \cdot 10^0$	32 sec $7,24 \cdot 10^{-14}$	40 sec $7,24 \cdot 10^{-14}$	63 sec $7,24 \cdot 10^{-14}$
35	32 sec $2,58 \cdot 10^0$	43 sec $2,60 \cdot 10^{-15}$	54 sec $2,06 \cdot 10^{-19}$	84 sec $2,06 \cdot 10^{-19}$
40	42 sec $2,58 \cdot 10^0$	55 sec $2,60 \cdot 10^{-15}$	70 sec $7,89 \cdot 10^{-23}$	108 sec $7,89 \cdot 10^{-23}$
⋮				
75	145 sec $2,58 \cdot 10^0$	182 sec $2,60 \cdot 10^{-15}$	249 sec $1,79 \cdot 10^{-47}$	390 sec $3,01 \cdot 10^{-54}$
80	164 sec $2,58 \cdot 10^0$	206 sec $2,60 \cdot 10^{-15}$	282 sec $1,79 \cdot 10^{-47}$	446 sec $2,11 \cdot 10^{-58}$

**2. táblázat.** Szükséges CPU idő (másodpercben) és a  $t = 50$  időpillanatban a függvényérték befoglalásának szélessége az  $\alpha = 1,5$  eset vizsgálatakor. A táblázat sorai az összes használt derivált maximális rendjét, míg az oszlopok a számítások során használt intervallumok végpontjainak pontosságát jelentik (C-XSC-ben).

Az eddigi eredményekből láthatjuk, hogy a használt lépésmagyság, a számítási pontosság és a maximális deriváltrend erősen befolyásolja a bizonyítás sikerességét és annak idejét. Ezen paraméterek optimális beállítása egy újabb probléma. Nézzük meg, hogyan alakul a trajektória befoglalásának szélessége. A 3. táblázat az adott időpillanatra a befoglaló intervallum szélességét tartalmazza. Láthatjuk, hogy egy egységnyi idő alatt a befoglalás szélessége körülbelül a kétszeresére nő. Ez az arány más  $\alpha$  értékekre is csak kicsit romlik. Közelítő számításokból tud-

juk, hogy a feltett kérdésre melyik lehet a megfelelő 1 hosszú szakasz. Ebből a két tulajdonságból már meghatározható, hogy hol milyen széles lehet a befoglaló intervallum egy sikeres bizonyítás esetén.

t	szélesség	t	szélesség	t	szélesség	t	szélesség
1	$9,12 \cdot 10^{-31}$	11	$1,97 \cdot 10^{-23}$	21	$8,66 \cdot 10^{-20}$	31	$1,88 \cdot 10^{-15}$
2	$2,26 \cdot 10^{-30}$	12	$9,33 \cdot 10^{-23}$	22	$2,06 \cdot 10^{-19}$	32	$4,60 \cdot 10^{-15}$
3	$2,71 \cdot 10^{-30}$	13	$2,30 \cdot 10^{-22}$	23	$4,59 \cdot 10^{-19}$	33	$1,02 \cdot 10^{-14}$
4	$3,37 \cdot 10^{-30}$	14	$6,29 \cdot 10^{-22}$	24	$8,43 \cdot 10^{-19}$	34	$2,32 \cdot 10^{-14}$
5	$3,89 \cdot 10^{-30}$	15	$1,35 \cdot 10^{-21}$	25	$1,64 \cdot 10^{-18}$	35	$5,31 \cdot 10^{-14}$
6	$4,24 \cdot 10^{-30}$	16	$2,30 \cdot 10^{-21}$	26	$6,93 \cdot 10^{-18}$	36	$1,05 \cdot 10^{-13}$
7	$4,90 \cdot 10^{-30}$	17	$4,58 \cdot 10^{-21}$	27	$1,16 \cdot 10^{-17}$	37	$2,05 \cdot 10^{-13}$
8	$7,33 \cdot 10^{-30}$	18	$1,13 \cdot 10^{-20}$	28	$3,46 \cdot 10^{-17}$	38	$4,49 \cdot 10^{-13}$
9	$2,85 \cdot 10^{-28}$	19	$2,49 \cdot 10^{-20}$	29	$2,22 \cdot 10^{-16}$	39	$9,95 \cdot 10^{-13}$
10	$2,24 \cdot 10^{-25}$	20	$4,42 \cdot 10^{-20}$	30	$6,38 \cdot 10^{-16}$	40	$1,98 \cdot 10^{-12}$

**3. táblázat.** A befoglalás szélességének alakulása a  $t$  időpillanatban az  $\alpha = 1,5$  esetben.

Az eddigi ismereteink alapján összeállítottunk egy optimalizáló eljárást [5] a paraméterek helyes beállítására. Az eljárás egy korlátos, három paraméteres egészértékű optimalizálási problémát old meg. A három paraméter a lépésmagyság, a műveletek pontossága és a maximális deriváltrend. Ezen paraméterek mellett követjük a trajektóriát egy adott konstans ideig. Az ehhez szükséges CPU időt fogjuk minimalizálni. A korlátozó feltétel pedig legyen az, hogy ebben az időpillanatban a befoglalás maximális szélessége ne legyen nagyobb, mint egy adott konstans.

Ez az eljárás az  $\alpha = 1,5$ -re a 20 időpillanatban a  $10^{-20}$ -os korlátozó feltétellel a végső befoglalás szélességére a 10,6 másodperces optimumot adta. Az optimális megoldás:

- Deriváltak száma: 28,
- Lépésmagyság:  $2^{-3}$ ,
- Számítás: 2 dupla pontos számmal.

Ez az eredmény közel azonos a korábban tapasztaltakkal. A magasabbrendű deriváltakra vonatkozó követelményt a korlát túl kicsire állításával lehet magyarázni. A  $10^{-20}$ -os korlátnál nagyobb is elegendő a bizonyításhoz. Az optimum értékéből a szükséges teljes időre a  $10,6 \cdot \frac{54}{20} = 28$  másodperces becslést kaphatjuk, mely úgyszintén igazodik az eddigi eredményeinkhez.



A következő lépésként egy  $\alpha$  intervallumon próbáltuk bizonyítani a feladat állítását. Mivel a kezdeti intervallumok szélessége erősen befolyásolja a későbbi befoglalások pontosságát, így széles intervallumokkal nem dolgozhatunk  $\alpha$  esetében. Az  $\alpha = [1, 5, 1, 5 + 10^{-22}]$  a legszélesebb olyan intervallum, melyre sikerült bizonyítani a feladat állítását. A futási paraméterek és a szükséges CPU idő hasonló volt, mint az előző esetben.

A harmadik kísérletben egy nagyobb  $\alpha$ -t választottunk,

$$\alpha = 1,546875 = 1 + \frac{1}{2} + \frac{1}{32} + \frac{1}{64},$$

amely pontosan ábrázolható számítógépen. Ehhez magasabbrendű deriváltakat is kellett használni.

Lépéshosszként  $2^{-4} = 1/16$ -ot használtunk. Ekkor a  $[115, 938, 116, 938]$  intervallum volt az első olyan 1 hosszú szakasz, mely teljesítette a feltételt. A feladat bonyolultságát jelzi, hogy ehhez már több órányi CPU időre volt szükség.

A továbbiakban több  $\alpha$  értékkel is futtattuk a programot. Minél nagyobb volt az  $\alpha$  értéke, azaz minél jobban közelítettünk a  $\pi/2$ -höz, annál nagyobb pontosságra volt szükségünk. Ennek következtében a szükséges deriváltak száma, valamint a számításhoz használt CPU idő is jelentősen megnőtt. Az ellenőrizhető  $\alpha$  intervallumok szélessége pedig erősen csökkent.

A programot több beállítással is futtattuk, egy a  $\pi/2$ -nél kicsit kisebb, számítógépen pontosan ábrázolható,  $\alpha$  értékkel is. Ezen tesztek eredményéből azt a következtetést vontuk le, hogy a számításokhoz szükséges pontosság legalább  $10^{-500}$ , és a szükséges CPU idő több mint 100 nap. Elméletileg a  $\pi/2$ -re is lehetne igazolni az állítást. A  $\pi/2$  környezetében megjelenő periodikus pályák szélsőértékei kicsik. Az ezekhez tartó megoldások esetében is létezik egy megfelelő 1 hosszú szakasz. Ebben az esetben a  $\pi/2$  értéket befoglaló intervallummal számolva igazolható a kérdés.

Összefoglalva, a feladat bizonyítása a teljes  $\alpha = [1, 5, \pi/2]$  intervallumra az óriási CPU idő miatt ezzel a módszerrel egyelőre nem lehetséges, bár a program az intervallum bármely pontjára, illetve annak szűk intervallumára képes matematikai bizonyítást adni.

## 8. Összefoglalás

Egy késleltetett differenciálegyenlet viselkedésének vizsgálatára adtunk megbízható módszert. Ezen eljárás alapja a szokásos differenciálegyenleteknél használt Taylor-sor egy megbízható formája.

A bizonyítás sikere minden esetben azon múlt, hogy mennyire pontosan számoltunk. Ennek két összetevője volt, az alkalmazott Taylor-polinom fokszáma, illetve a számítás pontossága. Mind a kettő erősen befolyásolta a számítási igényt is. Sikertelen volt a teljes állítás bizonyítása egy futáson belül.

A tapasztalatok alapján, ha  $\alpha$  tart a  $\pi/2$ -höz, akkor a probléma egyre nehezebb, de a trajektóriák viselkedése hasonló marad. Ha nincs az  $[1, 5, \pi/2]$  intervallumon belül szokatlanul viselkedő rész, akkor az intervallum bármely pontjára, illetve annak egy elegendően szűk intervallumára képesek vagyunk matematikai bizonyítást adni.

A jelen módszer alkalmazható lehet egyéb késleltetett differenciálegyenletek vizsgálatakor is.

## 9. Köszönetnyilvánítás

A szerző köszöni az Osztrák-Magyar (öu56011), valamint a Spanyol-Magyar (E-25/04 sz.) együttműködési pályázatok és az Országos Tudományos Kutatási Alapprogramok által finanszírozott (T 037491, T 048377) pályázatok támogatását.

## Hivatkozások

- [1] G. ALEFELD – J. HERZBERGER: *Introduction to Interval Computations*. Academic Press Inc., (1983)
- [2] B. BÁNHÉLYI – T. CSENDES: *A verified computational technique to locate chaotic regions of Hénon systems*. Proceedings of the 6th International Conference on Applied Informatics (konferenciaanyag), (2004), (297–304. p.)
- [3] B. BÁNHÉLYI – T. CSENDES – B. M. GARAY: *Optimization and the Miranda approach in detecting horseshoe-type chaos by computer*. International Journal of Bifurcation and Chaos (megjelenik 2007-ben)
- [4] M. BERZ – K. MAKINO – K. SHAMSEDDINE, – G. HOFFSTÄTTER – W. WAN: *Computational Differentiation: Techniques, Applications, and Tools*. COSY INFINITY and its Applications to Nonlinear Dynamics, SIAM, (1996), (365–367. p.)
- [5] T. CSENDES: *Nonlinear parameter estimation by global optimization - efficiency and reliability*. Acta Cybernetica **8**. (1988), (361–370. p.)
- [6] T. CSENDES – B. M. GARAY – B. BÁNHÉLYI: *A verified optimization technique to locate chaotic regions of a Hénon system*. Journal of Global Optimization **53**. (2006), (145–160. p.)
- [7] T. CSENDES – B. BÁNHÉLYI – L. HATVANI: *Towards a computer-assisted proof for chaos in a forced damped pendulum equation*. Journal of Computational and Applied Mathematics, **199**. (2007), (378–383. p.)
- [8] R. KLATTE – U. KULISCH – C. LAWO – M. RAUCH – A. WIETHOFF: *C-XSC: A C++ Class Library for Extended Scientific Computing*. Springer-Verlag, Berlin, (1993)

- [9] O. KNÜPEL: *A Multiple Precision Arithmetic for PROFIL*. Berichte des Forschungsschwerpunktes Informations- und Kommunikationstechnik, TU Hamburg-Harburg **93.6**, (1993)
- [10] O. KNÜPEL: *PROFIL – Programmers Runtime Optimized Fast Interval Library*. Berichte des Forschungsschwerpunktes Informations- und Kommunikationstechnik, TU Hamburg-Harburg, **93.4**, (1993)
- [11] T. KRISZTIN: *Periodic orbits and the global attractor for delayed monotone negative feedback*. Electronic Journal of Qualitative Theory of Differential Equations **15**, (2000), (1–12. p.)
- [12] R. E. MOORE: *Interval Analysis*. Prentice-Hall, (1966)
- [13] N. S. NEDIALKOV – K. R. JACKSON – G. F. CORLISS: *Validated Solutions of Initial Value Problems for Ordinary Differential Equations*. Applied Mathematics and Computation **105**, (1999), (21–68. p.)
- [14] E. M. WRIGHT: *A non-linear difference-differential equation*. Journal für die Reine und Angewandte Mathematik, **194**, (1955), (66–87. p.)

(Beérkezett: 2005. december 12.)

BÁNHÉLYI BALÁZS  
SZEGEDI TUDOMÁNYEGYETEM  
INFORMATIKAI TANSZÉKCSOPORT  
6720 SZEGED, ÁRPÁD TÉR 2.  
banhelyi@inf.u-szeged.hu

## DISCUSSION OF A DELAYED DIFFERENTIAL EQUATION WITH VERIFIED COMPUTING TECHNIQUE

BALÁZS BÁNHÉLYI

Consider the following delayed differential equation:

$$y' = -\alpha \left( e^{y(t-1)} - 1 \right),$$

where  $\alpha \in \mathbb{R}^+$  is a parameter. Let the initial function be

$$\phi(s) \equiv -11,$$

where  $s \in [-1, 0]$ .

When  $\alpha \leq 1,5$ , it is known, that the trajectory converges to zero, and when  $\alpha \geq \pi/2$ , the trajectory converges to different periodic solutions too. The analysis of the convergence to zero

*Alkalmazott Matematikai Lapok (2007)*

is very hard with numerical methods, so we consider an easier problem. We are interested in checking whether for all  $\alpha \in [\frac{3}{2}, \frac{\pi}{2}]$ , there exists a unit length time segment where the absolute value of the solution is less than 0,075.

Most verified techniques for solving ordinary differential equations apply a Taylor series. Our technique is based on the same idea too.

In this case the verification means mathematical verification, hence rounding and other errors were considered. Instead of real numbers, we can also calculate with intervals. In case the bounds of the result interval are not representable, then they are rounded outward. In this problem we used the Multiple Precision Interval Arithmetic libraries (C-XSC, PROFIL/BIAS).

We use two fix length lists to store the solution bounds. The first list contains the solution and the derivatives on time intervals, which cover the unit length time segment. The other list stores the solution and the derivatives in concrete time points. We calculate the new elements of the lists with earlier discussed formula. The oldest elements are deleted from the lists, and the new ones are inserted. This technique has three parameters: step length, maximum derivate rank, and a precision of the interval arithmetic. We combine our method and optimization technique to determine the optimal values for these parameters.

We proved the above statement for some tiny intervals around certain computer representable numbers, but we were not able to prove it for all points of the  $\alpha$  parameter interval.