

EÖTVÖS LORÁND TUDOMÁNYEGYETEM  
TERMÉSZETTUDOMÁNYI KAR

---

Nemecskó István

**INTERPOLÁCIÓS MÓDSZEREK ÉS ALKALMAZÁSUK  
DIFFERENCIÁLEGYENLETEK NUMERIKUS  
MEGOLDÁSÁRA**

BSc Szakdolgozat  
Matematika BSc, Matematikai elemző szakirány

Témavezető:

Csomós Petra

Alkalmazott Analízis és Számításmatematikai Tanszék



Budapest, 2017.

# Tartalomjegyzék

<b>Bevezetés</b>	<b>2</b>
<b>1. Interpolációs módszerek</b>	<b>5</b>
1.1. Lagrange interpoláció	6
1.1.1. Az interpolációs polinom Lagrange féle előállítás	6
1.2. Newton módszer	9
1.2.1. Osztott differenciák módszere	10
1.3. Az interpolációs hiba	13
1.4. Interpoláció Csebisev-féle alappontokon	17
1.5. Hermite interpoláció	19
1.6. Szakaszonként polinomiális interpoláció (Spline interpoláció)	21
<b>2. A közönséges differenciálegyenletek kezdetiérték feladatainak numerikus megoldása</b>	<b>25</b>
2.1. A közönséges differenciál egyenletek kezdetiértékfeladata	25
2.2. Egylépéses módszerek	27
2.3. Explicit Euler módszer	28
2.4. Az Explicit Euler-módszer hibafüggvénye	28
2.5. Az Explicit Euler módszer konvergenciája	29
2.6. Implicit Euler-módszer	31
2.7. A trapéz módszer	32
2.8. A $\Theta$ -módszer	32
2.9. Magasabb rendű módszerek	33
2.10. Runge–Kutta módszerek	34
<b>3. Interpolációs módszerek alkalmazása differenciálegyenletek numerikus megoldására</b>	<b>35</b>
3.1. A harmonikus rezgőmozgás egyenlete	35
3.2. Explicit Euler-módszer alkalmazása	36
3.3. Lagrange interpoláció alkalmazása a numerikus megoldásra	37
3.4. Lagrange interpoláció Csebisev alappontokon	39
3.5. Spline interpoláció	39
<b>Összefoglalás</b>	<b>41</b>
<b>Függelék</b>	<b>43</b>

## Köszönetnyilvánítás

Elsősorban szeretném megköszönni Csomós Petra témavezetőmnek a rengeteg segítséget, amit a dolgozat megírása közben kaptam tőle. Köszönöm a sok konzultációs lehetőséget, azt, hogy minden kérdésre legnagyobb körültekintéssel válaszolt, a dolgozatomat rengetegszer átnézte.

Szeretném még megköszönni családomnak, hogy tanulmányaim során végig támogattak.

Hálával tartozom Faragó István professzor úrnak, hogy az egyetemi tanórákat olyan érdekesen tartotta, hogy az egész témakört megkedveltette velem.

# Bevezetés

A dolgozatban numerikus módszereket vizsgálunk, amelyek alkalmazhatóságát nézzük egy ismert tesztfeladat megoldására. A harmonikus rezgőmozgást leíró differenciálegyenletet fogjuk numerikus módszerekkel megoldani, majd a megoldás ábrázolására is numerikus módszereket alkalmazunk. Ezeket a módszereket fogjuk összehasonlítani. A MATLAB program segítségével könnyen kirajzolhatjuk a pontos megoldást, amit közelíteni szeretnénk. Felmerülhet a kérdés, hogy a MATLAB milyen beépített módszereket használ a differenciálegyenlet megoldására, illetve annak kirajzolását milyen interpolációs eljárással végzi?

A természetbeli, műszaki vagy gazdasági folyamatok leírása gyakran differenciálegyenletekkel lehetséges. A differenciálegyenletek gyakran elemi módszerekkel nem oldhatók meg, ezért alkalmazunk numerikus módszereket a megoldás előállítására. A dolgozat második fejezetében fogunk foglalkozni a differenciálegyenletek numerikus megoldásával. A fejezetben különböző módszereket vizsgálunk a megoldás előállítására. Az eredeti megoldást minél pontosabban szeretnénk közelíteni, ezért igyekszünk nem csak elsőrendű módszert adni a numerikus megoldás előállítására. Amennyiben a differenciálegyenletet megoldjuk numerikusan, úgy a megoldás pontpárok halmaza lesz. Ezen pontokat összekötve szeretnénk a pontos megoldás függvény minél pontosabb közelítését kapni. A pontok összekötését is megoldhatjuk numerikus módszerek segítségével.

A dolgozatban a pontok összekötésére interpolációs módszereket alkalmazunk. Az első fejezetben interpolációs módszerekről lesz szó. Az interpoláció egy közelítő eljárás, amelynek segítségével egy függvény ismert értékeiből az ismeretlen, közttes értékekre adhatunk becslést. Az első fejezetben vizsgáljuk a módszerek előnyeit és hátrányait is alkalmazhatóság szempontjából.

Célunk tehát, hogy a harmonikus rezgőmozgás ismert, pontos megoldását minél jobban közelítsük numerikus módszerekkel, ezáltal vizsgálva ezek pontosságát. A dolgozat harmadik fejezetében fogunk foglalkozni a harmonikus rezgőmozgás megoldását minél jobban közelítő numerikus módszer megtalálásával. A harmonikus rezgőmozgás egyenletét leíró differenciálegyenletet megoldjuk az Explicit Euler és az Implicit Euler módszer segítségével. A numerikus megoldására saját fejlesztésű MATLAB programkódot használunk. A kódok csatolva megtalálhatóak a függelékek szakaszban.

A differenciálegyenlet numerikus megoldásaira alkalmazunk interpolációs eljárásokat. Vizsgálni fogjuk, hogy az interpolációs polinom Lagrange féle előállítása milyen pontosan közelíti az eredeti megoldásfüggvényt. Továbbá a megoldás minél pontosabb előállítása érdekében vizsgáljuk a Csebishev alappontokon történő interpolációt is. Végül a pontosság növelése érdekében alkalmazzuk a Spline interpolációt is.

A Lagrange interpolációra saját fejlesztésű MATLAB programot fogunk használni. A program kódja megtalálható a függelékek részben.

# 1. fejezet

## Interpolációs módszerek

Az interpoláció egy közelítő eljárás, amely egy függvény ismeretlen értékeire ad választ az ismert értékekből, azaz az ismert függvényértékek közötti ismeretlen értékekre ad becslést. Tehát egy derékszögű koordináta rendszer pontjait szeretnénk összekötni polinomok, vagy trigonometrikus polinomok grafikonjaival. Ezáltal az ismert pontokból egy adott eljárással az ismeretlen pontokat is megkaphatjuk. Ez az eljárás számos területen alkalmazható. Például:

- Olajkút tervezésekor megvan már néhány mérési eredmény, hogy milyen mélyre kell fúrni az olajhoz, azokból becsülhető a köztes helyeken is a kellő fúrési mélység.
- Egy héten át minden nap megmérjük a hőmérsékletet reggel, délben és este, ezekből az adatokból szeretnénk a köztes időpontokban meghatározni a hőmérsékletet.
- A GPS műhold csak bizonyos időközönként küld adatot a helyzetéről, viszont fontos tudni, hogy az adott időpontok között pontosan hol tartózkodott.

Az interpoláció célja tehát az adott pontokon átmenő polinom, polinomok meghatározása. Erre többféle eljárást ismerünk: az interpolációs polinom Lagrange, illetve Newton-féle előállítás, Hermite interpoláció, ezen belül a Hermite–Fejér interpoláció, illetve Spline interpoláció. Továbbá célunk a legpontosabb megoldás meghatározása, ezért azt is vizsgálunk kell, hogy az alappontok megválasztása hogyan befolyásolja az interpoláció hibáját.

Jelen dolgozatban a Csebisev alappontokon történő interpolációt fogjuk vizsgálni.

A dolgozatban Faragó I. és Horváth R. [2], J. R. Chasnov [1], J. Stoer és R. Bulirsch [3], T. Young és M. J. Mohlenkamp [4] jegyzeteket interpolációval és közönséges differenciálegyenletek numerikus megoldásával foglalkozó fejezeteit dolgoztuk fel.

Tegyük fel, hogy  $f : \mathbb{R} \rightarrow \mathbb{R}$  függvény. Tegyük fel, hogy ennek a függvénynek ismerjük  $n + 1$  helyen az értékét. Ekkor ezek a pontok legyenek:  $(x_i, f_i)$ , ahol  $i = 0, \dots, n$ , ahol feltesszük, hogy  $x_i \neq x_j$ , ha  $i \neq j$ . Ekkor  $(x_i)$ -ket alappontoknak nevezzük. Legyen  $I = [x_{min}, x_{max}]$ , ahol  $x_{min} = \min(x_0, \dots, x_n)$ ,  $x_{max} = \max(x_0, \dots, x_n)$ . A célunk, hogy tudjunk becslést adni  $f$  függvény értékeire  $I$  intervallumon belüli, alappontoktól különböző pontokra is. Továbbá, hogy megbecsüljük  $f$  függvény deriváltjait, integrálját, szélsőértékeinek helyét.

Legyen  $\mathbb{P}_n$  a legfeljebb  $n$ -ed fokú polinomok halmaza. Ekkor keressük  $g \in \mathbb{P}_n$ -t, ami a legjobban közelíti  $f$ -et.

Megjegyzés: Egy  $n$ -ed fokú polinomnak  $n + 1$  darab együtthatóját választhatjuk meg szabadon, így  $n + 1$  ponthoz egy legfeljebb  $n$ -ed fokú polinomra van szükségünk, így keressük  $g \in \mathbb{P}_n$  függvényt.

Mostantól csak olyan közelítésekkel foglalkozunk, amelyek teljesítik az interpolációs tulajdonságot, azaz a keresett  $g$  függvény áthalad az adott  $f$  függvény adott értékein, tehát  $g(x_i) = f(x_i) = f_i$   $i = 0, \dots, n$ -re.

A következő fejezetekben különböző eljárásokat vizsgálunk az interpolációs polinom előállítására. Foglalkozunk az interpolációs polinom Lagrange-féle előállításával, a Newton-féle előállításával,

a Hermite-féle interpolációval, illetve Spline interpolációval. Vizsgálni fogjuk a különböző módszerek előnyeit, hátrányait, illetve az interpolációs hibát, illetve, hogy az alappontok megválasztása hogyan befolyásolja az interpolációs hibát.

## 1.1. Lagrange interpoláció

Legyen az  $(x_i, f_i)$ ,  $i = 0, \dots, n$  az interpoláció alapjául szolgáló adatok  $(x_i \neq x_j, i \neq j)$ . Keressük azt a  $g \in \mathbb{P}_n$  polinomot, amelyre az alábbi, úgynevezett interpolációs feltétel teljesül:

$$g(x_i) = f_i. \quad (1.1)$$

Az (1.1) feladatot interpolációs alapfeladatnak nevezzük.

**1.1.1. Tétel (Egyértelműség).** *Ha az (1.1) interpolációs feladatnak létezik megoldása, akkor az egyértelmű. Azaz egyértelműen létezik  $g \in \mathbb{P}_n$  polinom, amelyre az 1.1 interpolációs tulajdonság teljesül.*

*Bizonyítás.* 1: A bizonyítás módszere indirekt. Tegyük fel, hogy létezik  $g, h \in \mathbb{P}_n$  különböző polinomok amelyek megoldásai (1)-nek. Ekkor tudjuk, hogy  $g - h$  szintén  $\in \mathbb{P}_n$ . Mivel  $g(x_i) = f_i = h(x_i)$  definíció szerint, ezért  $p(x_i) = g(x_i) - h(x_i) = f_i - f_i = 0$  Tehát a  $p$  polinomnak minden  $x_i$  gyöke ( $i = 0, \dots, n$ ) esetén. Az algebra alaptétele szerint egy legfeljebb  $n$ -ed fokú polinomnak legfeljebb  $n$  darab különböző valós gyöke lehet. Tehát  $p \in \mathbb{P}_n$ -nek csak akkor lehet  $n + 1$  különböző gyöke ha  $p \equiv 0$  polinom. Ekkor viszont  $p = g - h = 0$  egyenlőségből követkeik, hogy  $g = h$  tehát ellentmondásra jutottunk, mert  $g$  és  $h$  azonosak.

Megjegyzés: Ezt az egyértelmű  $p$  polinomot jelöljük  $L_n$  -nel. □

*Bizonyítás.* 2: Keressük az  $L_n$  polinomot

$$L_n(x) = \sum_{k=0}^n a_k \cdot x^k$$

alakban, ahol  $a_k \in \mathbb{R}$ ,  $\forall k = 0 \dots n$ . Ekkor az interpoláció feltétele a következő:  $L_n(x_i) = \sum_{k=0}^n a_k \cdot x_i^k = f_i$  ( $i = 0, \dots, n$ )-re. Ez egy olyan lineáris egyenletrendszer (LAER) aminek az együttható mátrixa egy *Vandermonde*-mátrix, és mivel az interpoláció feltétele, hogy  $x_i \neq x_j, i \neq j$  ezért a mátrix determinánsa nem 0. Ebből következik, hogy a LAER egyértelműen megoldható, azaz az  $a_k$  együtthatók egyértelműen meghatározhatóak. Ezzel beláttuk az állítást. □

### 1.1.1.1. Az interpolációs polinom Lagrange féle előállítás

**1.1.2. Tétel (Unicitás).** *Mutassuk meg, hogy az (1.1) feladatnak létezik megoldása:*

*Bizonyítás.* Legyen

$$\ell'_i(x) = \prod_{j=0; j \neq i}^n (x - x_j)$$

Ekkor  $\ell'_i \in \mathbb{P}_n$ , továbbá bármelyik  $x_j$  alappontot behelyettesítve  $\ell'_i$ -be 0-t kapunk, ha  $i \neq j$  és csak akkor kapunk nem 0-t, ha  $i = j$ . Ekkor:

$$\ell_i(x) = \frac{\ell'_i(x)}{\ell'_i(x_i)} = \frac{\prod_{j=0; j \neq i}^n (x - x_j)}{\prod_{j=0; j \neq i}^n (x_i - x_j)} = \prod_{j=0; j \neq i}^n \frac{(x - x_j)}{(x_i - x_j)}.$$

Ekkor  $\ell_i(x)$  szintén  $\mathbb{P}_n$ -beli továbbá  $\ell_i(x_j) = 0$ , ha  $i \neq j$  és  $\ell_i(x_j) = 1$ , ha  $i = j$  Az  $\ell_i(x)$ -et az  $i$ -edik alapponthoz tartozó Lagranfe-féle alappolinomnak nevezzük.

Legyen  $L_n(x) = \sum_{i=0}^n f_i \cdot \ell_i(x)$  ekkor  $L_n \in \mathbb{P}_n$ , mert  $\ell_i(x)$ -ek összegeként áll elő, amik szintén  $\mathbb{P}_n$ -beliek.

Vizsgáljuk meg, hogyan viselkedik  $L_n$  az alappontokban:

$$L_n(x_j) = \sum_{i=0}^n f_i \cdot \ell_i(x_j)$$

mivel  $\ell_i(x_j) = 0$  minden  $i \neq j$  esetén és  $\ell_i(x_j) = 1$  ha  $i = j$ , tehát a szummában csak az az összeadandó tag nem 0, ahol  $i = j$ , ebből következik, hogy

$$L_n(x_j) = \sum_{i=0}^n f_i \cdot \ell_i(x_j) = f_j, \quad j = 0, \dots, n$$

Tehát az így konstruált  $L_n$  polinom kielégíti az (1.1) feladat feltételét.

**1.1.1. Definíció.** Az

$$L_n(x) = \sum_{i=0}^n f_i \cdot l_i(x) = \sum_{i=0}^n f_i \cdot \prod_{j=0, j \neq i}^n \frac{x - x_j}{x_i - x_j}$$

$\mathbb{P}_n$ -beli polinomot az (1) feladathoz tartozó Lagrange-féle interpolációs polinomnak nevezzük.

Legyen

$$\omega_{n+1}(x) = \prod_{j=0}^n (x - x_j) \tag{1.2}$$

Alkalmazva a szorzat deriváltjára vonatkozó összefüggést:

$$\omega'_{n+1} = \sum_{k=0}^n \prod_{j=0, j \neq k}^n (x - x_j),$$

másrészt

$$\omega'_{n+1}(x_i) = \prod_{j=0, j \neq i}^n (x_i - x_j).$$

Ebből következik, hogy

$$l_i(x) = \frac{\omega_{n+1}(x)}{(x - x_i) \cdot \omega'_{n+1}(x_i)}, \quad i = 0, \dots, n$$

• **Megjegyzés:**

Az az (1.1) feladat megoldása direkt módszerrel is megadható: keressük

$$p_n(x) = \sum_{i=0}^n a_i \cdot x^i$$

alakban a polinomot. Ekkor  $P_n \in P_n$  definíció szerint. Vizsgáljuk meg  $p_n(x)$ -et az alappontokban:

$$p_n(x_i) = \sum_{i=0}^n a_i \cdot x_i^i = f_i,$$

Vegyük észre, hogy  $p_n(x)$  teljesíti az interpolációs feltételt.

Ekkor az együtthatókból alkotott vektor legyen  $(\vec{a} = [a_0, a_1, \dots, a_n])$  és hasonlóan  $\vec{f} = [f_0, f_1, \dots, f_n]$ . Továbbá  $B \in \mathbb{R}^{(n+1) \times (n+1)}$  mátrix legyen a következő:

$$B = \begin{pmatrix} 1 & x_0 & x_0^2 & \dots & x_0^n \\ 1 & x_1 & x_1^2 & \dots & x_1^n \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_n & x_n^2 & \dots & x_n^n \end{pmatrix}$$

Ekkor a következő lineáris egyenletrendszer kell megoldanunk:  $B \cdot \vec{a} = \vec{f}$ . Mivel  $B$  egy Vandermonde mátrix, ennek a determinánsa

$\prod_{0 \leq j < i \leq n} (x_i - x_j)$  amiről tudjuk, hogy nem nulla, hiszen alapfeltétel volt, hogy  $x_i \neq x_j$ , ha  $i \neq j$ . Mivel a determináns nem 0 ezért egyértelműen létezik megoldása az (1) feladatnak. Ezt kellett belátnunk.

□

1.1.1. *Példa megoldás.* Határozzuk meg, az  $(1, 5), (3, 2), (4, 3)$  pontokra illeszkedő interpolációs polinomot a Lagrange módszerrel. Alkalmazva a Lagrange-interpoláció jelöléseit ebben az esetben  $n = 2$  és  $x_0 = 1, x_1 = 3, x_2 = 4$  és  $f_0 = 5, f_1 = 2, f_2 = 3$ .

Első lépésben határozzuk meg  $\ell_0(x), \ell_1(x), \ell_2(x)$  polinomokat. Alkalmazzuk a következő összefüggést:

$$\ell'_i(x) = \prod_{j=0; j \neq i}^n (x - x_j), \quad i = 0, \dots, n$$

Ekkor az  $x_0, x_1, x_2$  pontokhoz tartozó alappolinomok:

$$\begin{aligned} \ell_0(x) &= \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)} = \\ &= \frac{(x - 3)(x - 4)}{(1 - 3)(1 - 4)} = \frac{x^2 - 7x + 12}{6}. \\ \ell_1(x) &= \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)} = \\ &= \frac{(x - 1)(x - 4)}{(3 - 1)(3 - 4)} = \frac{x^2 - 5x + 4}{-2}. \\ \ell_2(x) &= \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)} = \\ &= \frac{(x - 1)(x - 3)}{(4 - 1)(4 - 3)} = \frac{x^2 - 4x + 3}{3}. \end{aligned}$$

Alkalmazva az alábbi összefüggést:

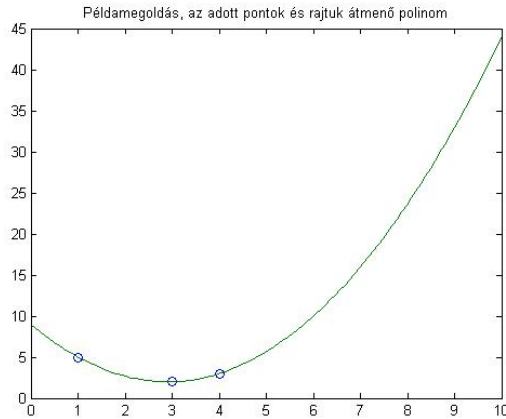
$$L_n(x) = \sum_{i=0}^n f_i \cdot \ell_i(x)$$

meghatározhatjuk  $L_2(x)$ -t, azaz az adott három alapponthoz tartozó Lagrange-féle interpolációs polinomot. Tehát

$$\begin{aligned} L_2(x) &= \frac{x^2 - 7x + 12}{6} \cdot 5 + \frac{x^2 - 5x + 4}{-2} \cdot 2 + \frac{x^2 - 4x + 3}{3} \cdot 3 = \\ &= \frac{1}{6} \cdot (5x^2 - 29x + 54) \end{aligned}$$

Oldjuk meg a programmal is a feladatot, ábrázoljuk a megoldást. A programot én készítettem, a fent alkalmazott képletekkel számol, és azok alapján készíti el az interpolációs polinomot.





1.1. ábra.

Az 1.1. ábrán láthatjuk folytonos zöld vonallal az Lagrange-féle interpolációs polinomot, illetve kék körökkel az adott pontokat. Látható, hogy az interpolációs polinom áthalad az adott pontokon, ahogy azt elvárjuk.

- **Összegzés:** Ismert, hogy a Lagrange-féle interpolációs polinom egy adott helyen vett helyettesítési értéke  $4n^2 + O(n)$  művelettel számítható ki. Az eljárás előnye az, hogy ha valamelyik alappontban megváltoztatjuk a függvény értékét, akkor az új polinom helyettesítési értékeit könnyen újraszámolhatjuk a régi értékekből. Például, ha egy rosszul mért adat miatt egy adott alappontban új értéket kell felvenni. Hátránya azonban az eljárásnak, hogy egy új alappont felvétele esetén az összes eddigi értéket újra kell számolni. Például, ha egy új mért értékkel egészítjük ki az eddigi adatsort. Ezen probléma kiküszöbölésére vizsgáljunk egy újabb módszert az interpolációs polinom kiszámolására. Ez a Newton-módszer.

## 1.2. Newton módszer

Az előző fejezetben beláttuk, hogy az interpolációs polinom egyértelmű, azaz egyetlen olyan polinom létezik, amely kielégíti a megadott feltételeket. Tehát a következőkben ennek a polinomnak egy másik fajta előállítását fogjuk vizsgálni. Jelölje most ezt a polinomot  $N_n(x)$ . Ekkor  $N_n(x)$  azt a legfeljebb  $n$ -ed fokú polinomot jelöli, amire  $N_n(x_i) = f_i$  ( $i = 0, \dots, n$ ) esetén, azaz teljesül rá a az (1.1) interpolációs feltétel. Ekkor  $N_n(x)$ -et az  $(x_i; f_i)$   $i = 0, \dots, n$  alappontokhoz tartozó Newton féle interpolációs polinomnak nevezzük. Tehát  $N_n(x) = L_n(x)$ .

- A polinom előállítása: Keressük a polinomot a következő alakban:

$$\begin{aligned}
 N_n(x) = & b_0 + b_1 \cdot (x - x_0) + b_2 \cdot (x - x_0) \cdot (x - x_1) + \\
 & + \dots + \\
 & + b_n \cdot (x - x_0) \cdot (x - x_1) \cdot \dots \cdot (x - x_{n-1}).
 \end{aligned}$$

Mivel az egyes összeadandó tagok lineárisan függetlenek, ezért egyértelműen léteznek olyan  $(b_0, \dots, b_n)$  konstansok amik megválasztásával az interpolációs polinomot kapjuk. Alkalmazzuk az előző részben bevezetett  $\omega$  függvényt, így átirva az előző összefüggést a következőt kapjuk:

$$\sum_{i=0}^n b_i \cdot \omega_i(x).$$

- Megjegyzés: Mivel  $\omega_0(x) = 1$  ezért  $b_0 = f_0$  és  $N_0(x) = \omega_0(x) \cdot b_0$ . Ennek a segítségével rekurzióval definiálhatjuk  $N_n(x_0)$ -et. Vegyük észre, hogy az  $N_n(x) - N_{n-1}(x) \in \mathbb{P}_n$ , továbbá  $N_n(x_i) - N_{n-1}(x_i) = 0$  minden  $i = 0, \dots, n-1$  esetén, az interpolációs alapfeltétel miatt. Tehát a fent említett polinomnak a gyökei:  $x_0, \dots, x_{n-1}$ . Amik pont a  $b_k \cdot \omega_n(x)$ -ek egyenlők. Tehát a következő rekurziós összefüggést kapjuk:

$$N_n(x) = N_{n-1}(x) + b_n \cdot \omega_n(x)$$

Az  $\sum_{i=0}^n b_i \cdot \omega_i(x)$  képletből tehát a  $b_i$  együtthatókat kell már csak meghatározni. Ehhez használjuk fel, hogy a Newton-módszerrel előállított polinom megegyezik a Lagrange-módszerrel előállított polinommal.

Tehát  $L_n(x) = N_n(x)$  és  $N_0(x) = f_0 = b_0 \cdot \omega_0(x)$ , ahol  $\omega_0(x) = 1$ , ezért  $b_0 = f_0$ . Vizsgáljuk meg  $L_n$ -ben  $x^n$  együtthatóját.

$$L_n(x) = \sum_{i=0}^n f_i \cdot \prod_{j=0, j \neq i}^n \frac{x - x_j}{x_i - x_j}.$$

Ebből következik, hogy  $x^n$  együtthatója:  $\sum_{i=0}^n f_i \cdot \prod_{j=0, j \neq i}^n \frac{1}{x_i - x_j}$ , mert  $n$ -szer szorozzuk össze  $x - x_i$ -t  $i = 0, \dots, n$  esetén.

Viszont a Newton-féle polinom  $N_n(x) = N_{n-1}(x) + b_n \cdot \omega_n(x)$  előállításából  $x^n$  együtthatója  $b_n$ , mert  $\omega_n(x)$ -ben szerepel csak  $x^n$ , és ennek együtthatója 1, és ezt szorozzuk  $b_n$ -el.

Mivel a két módszerrel előállított polinom megegyezik, ezért  $x^n$  együtthatójának is egyenlőnek kell lennie, tehát

$$b_n = \sum_{i=0}^n f_i \cdot \prod_{j=0, j \neq i}^n \frac{1}{x_i - x_j}.$$

- Megjegyzés: A képletbe behelyettesítve  $b_1 = \frac{f_1 - f_0}{x_1 - x_0}$

### 1.2.1. Osztott differenciák módszere

A következő részben a Newton-féle módszerrel előállított interpolációs polinom egy átláthatóbb felírási módját fogjuk vizsgálni.

**1.2.1. Definíció.** Legyen  $(x_i, f_i), i = 1, \dots, s$  ( $x_i \neq x_j$  ha  $i \neq j$ ) adott pontok. Ekkor az

$$[x_1 x_2 \dots x_s]f = \sum_{i=1}^s \frac{f_i}{(y_i - y_1) \cdot \dots \cdot (y_i - y_{i-1}) \cdot (y_i - y_{i+1}) \cdot \dots \cdot (y_i - y_s)}$$

számot az adott pontokhoz tartozó  $s - a - 1$ -ed rendű Newton féle osztott differenciának nevezzük.

Megjegyzés: A 0-ad rendű osztott differenciát  $[x_i]f = f_i$ -ként értelmezzük.

**1.2.1. Tétel.** Az osztott differenciák kiszámítására alkalmazhatjuk a következő képletet is:

$$[x_1 x_2 \dots x_s]f = \frac{[x_1 x_2 \dots x_s]f - [x_0 x_1 \dots x_{s-1}]f}{x_s - x_0}$$

*Bizonyítás.* :Lássuk be az 1.2.1 tételt:

$$[x_1 x_2 \dots x_n]f = \frac{[x_1 x_2 \dots x_n]f - [x_0 x_1 \dots x_{n-1}]f}{x_n - x_0} \rightarrow$$

$$[x_1 x_2 \dots x_n]f \cdot (x_n - x_0) = [x_1 x_2 \dots x_n]f - [x_0 x_1 \dots x_{n-1}]f$$

induljunk ki a jobboldalból és alakítsuk a definíció szerint.

$$\sum_{i=1}^n \frac{f_i \cdot (x_i - x_0)}{(x_i - x_0) \cdot (x_i - x_1) \cdots (x_i - x_{i-1}) \cdot (x_i - x_{i+1}) \cdots (x_i - x_n)} - \sum_{i=0}^{n-1} \frac{f_i \cdot (x_i - x_n)}{(x_i - x_n) \cdot (x_i - x_1) \cdots (x_i - x_{i-1}) \cdot (x_i - x_{i+1}) \cdots (x_i - x_{n-1})} =$$

Vegyük az első szummából az  $n$ -edik összeadandót, míg a második szummából az első összeadandót külön, ekkor a következőt kapjuk:

$$= \frac{f_n \cdot (x_n - x_0)}{(x_n - x_0) \cdots (x_n - x_{n-1})} - \frac{f_0 \cdot (x_0 - x_n)}{(x_0 - x_n) \cdots (x_0 - x_1)} + \sum_{i=1}^{n-1} \frac{f_i \cdot (x_n - x_0)}{(x_i - x_0) \cdot (x_i - x_1) \cdots (x_i - x_{i-1}) \cdot (x_i - x_{i+1}) \cdots (x_i - x_n)}.$$

Vegyük észre, hogy

$$\frac{f_0 \cdot (x_0 - x_n)}{(x_0 - x_n) \cdots (x_0 - x_n)} = - \frac{f_0 \cdot (x_n - x_0)}{(x_0 - x_n) \cdots (x_0 - x_n)}$$

Ezt kihasználva azt kapjuk, hogy

$$b_n = \sum_{i=0}^n \frac{f_i}{(x_i - x_0) \cdot (x_i - x_1) \cdots (x_i - x_{i-1}) \cdot (x_i - x_{i+1}) \cdots (x_i - x_n)} = [x_0 \dots x_n]f$$

Tehát beláttuk az állítást. □

**1.2.2. Tétel.** A  $b_i$  együtthatókat kiszámíthatjuk a következő képpen:

$$b_i = [x_0 \dots x_i]f, i = 0, \dots, n$$

képlet segítségével.

*Bizonyítás.*

$$b_n = \sum_{i=0}^n f_i \cdot \prod_{j=0, j \neq i}^n \frac{1}{x_i - x_j}$$

Vizsgáljuk meg, hogy a kétféle definíció  $b_n$ -re megegyezik:

Vegyük az adott  $x_i, i = 0, \dots, n$  alappontokhoz tartozó egyértelmű interpolációs polinomot, jelöljük  $p_n(x)$ -el

A Newton módszer definíciója szerint:  $p_n(x) = \sum_{i=0}^n b_i \cdot \omega_i(x)$

Mivel az interpolációs polinom egyértelmű, ezért írjuk fel a Lagrange módszerrel is a polinomot:

$$p_n(x) = \sum_{i=0}^n f_i \cdot \frac{(x - x_0)(x - x_1) \cdots (x - x_{i-1})(x - x_{i+1}) \cdots (x - x_n)}{(x_i - x_0)(x_i - x_1) \cdots (x_i - x_{i-1})(x_i - x_{i+1}) \cdots (x_i - x_n)}$$

Vegyük észre, hogy ennek a polinomnak a főegyütthatója

$$\sum_{i=0}^n \frac{f_i}{(x_i - x_0)(x_i - x_1) \cdots (x_i - x_{i-1})(x_i - x_{i+1}) \cdots (x_i - x_n)}$$

Tehát a Newton módszerrel előállított polinom főegyütthatójának (ez pont  $b_n$ ) is ennyinek kell lenni. Ez viszont definíció szerint az osztott differencia, tehát  $[x_0 \dots x_n]f$ . Ezzel beláttuk az állítást. □

1.2.1. *Példa megoldás.* Oldjuk meg a 1.0 fejezetben ismertett példafeladatot Newton módszerrel is. Tehát  $n = 2$ ,  $x_0 = 1$ ,  $x_1 = 3$ ,  $x_2 = 4$  és  $f_0 = 5$ ,  $f_1 = 2$ ,  $f_2 = 3$ . Ekkor

$$N_2(x) = b_0 + b_1(x - x_0) + b_2(x - x_0)(x - x_1)$$

képletet alkalmazva a következőt kapjuk:

$$N_2(x) = b_0 + b_1(x - 1) + b_2(x - 1)(x - 3).$$

Határozzuk meg  $b_0, b_1, b_2$  együtthatókat. Definíció szerint  $b_0 = f_0$ , továbbá  $b_1 = [x_1, x_0]f = \frac{f_1 - f_0}{x_1 - x_0}$  Newton-féle osztott differencia ezért  $b_1 = \frac{2-5}{3-1} = -\frac{3}{2}$ . Hasonlóan számítható ki  $b_2$  is:  $b_2 = [x_2, x_1, x_0]f = \frac{[x_2, x_1]f - [x_1, x_0]f}{x_2 - x_0}$ . Számítsuk ki még  $[x_2, x_1]f$ -et.  $[x_2, x_1]f = \frac{f_2 - f_1}{x_2 - x_1} = \frac{3-2}{4-3} = 1$ , ennek segítségével már  $b_2$  is kiszámítható.

Ezt behelyettesítve tehát  $b_2$  képletébe a következőt kapjuk:

$$b_2 = \frac{1 + \frac{3}{2}}{4 - 1} = \frac{5}{6}.$$

Mivel meghatároztuk  $b_0, b_1, b_2$  együtthatókat, helyettesítsünk be:

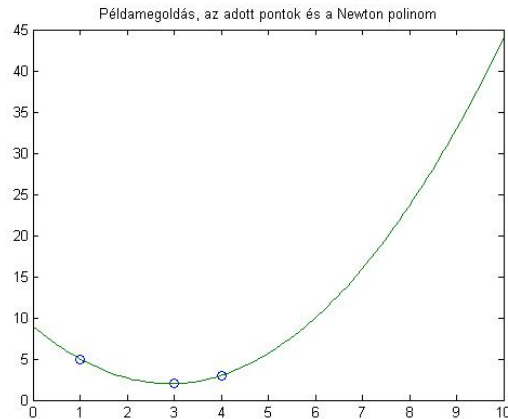
$$N_2(x) = 5 - \frac{3}{2}(x - 1) + \frac{5}{6}(x - 1)(x - 3) = \frac{1}{6}(5x^2 - 29x + 54).$$

Megjegyzés: Mivel az interpolációs polinom egyértelmű, ezért ugyanazt a polinomot kaptuk, mint amit a Lagrange módszerrel kaptunk.

Megjegyzés: A megoldás átláthatóságát könnyíthetjük, ha az osztott differenciákat a következő táblázat segítségével számoljuk:

$x_i$	$f_i = [.]f$	$[.,.]f$	$[.,.,.]f$
1	$5 = b_0$		
		$\frac{2-5}{3-1} = -\frac{3}{2} = b_1$	
3	2		$\frac{1+\frac{3}{2}}{4-1} = b_2$
		$\frac{3-2}{4-3} = 1$	
4	3		

Oldjuk meg a feladatot program segítségével, és ábrázoljuk a megoldást:



1.2. ábra.

Az ábrán zöld folytonos vonallal láthatjuk a Newton-féle interpolációs polinomot, illetve kék körrel jelölve az adott pontokat. Vegyük észre, hogy a polinom áthalad az adott pontokon, azaz teljesíti az (1.1) interpolációs alapfeladatot. Hasonlítsuk össze az 1.1 ábrával. Mivel az interpolációs polinom egyértelmű, ezért a két függvény megegyezik, ahogy az ábra is mutatja.

### 1.3. Az interpolációs hiba

Jelöljük  $p_n \in \mathbb{P}_n$ -el az interpolációs polinomot. Azaz teljesíti az alapfeltételt:  $p_n(x_i) = f(x_i), i = 0, \dots, n$ . Az  $f$  ismeretlen függvénnyel egyező értékeket kell felvegyen az adott alappontokban.

- Feladat: Adjunk becslést az

$$f(x) - p_n(x)$$

hibára, ha  $x \in [a, b]$ .

**1.3.1. Tétel.** Tegyük fel, hogy  $f \in C^{n+1}([a, b])$ ,  $n + 1$ -szer di erenciálható függvény. Ekkor létezik  $\xi \in (a, b)$  úgy, hogy

$$f(x) - p_n(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} \cdot \omega_{n+1}(x)$$

- Megjegyzés:  $\omega_{n+1}(x)$  az előző részben definiált  $\prod_{i=0}^n (x - x_i)$  függvény.
- Legyen  $x' \in [a, b]$  amely nem interpolációs alappont, azaz  $x' \neq x_i, i = 0, \dots, n$ .
- Megjegyzés: Mivel  $p_n(x_i) = f(x_i), i = 0, \dots, n$ -re, ezért az  $x_i, i = 0, \dots, n$  alappontokban a hiba 0.  $p_n(x_i) - f(x_i) = f(x_i) - f(x_i) = 0, i = 0, \dots, n$ .
- Legyen:  $\phi(x) = f(x) - p_n(x) - K\omega_{n+1}(x)$ , ahol  $K$  egy állandó. Legyen  $K$  olyan, hogy  $\phi(x') = 0$ . Azaz  $\phi(x') = f(x') - p_n(x') - K\omega_{n+1}(x')$ , Ezt átrendezve megkaphatjuk  $K$  értékét:

$$K = \frac{f(x') - p_n(x')}{\omega_{n+1}(x')}.$$

Ilyen  $K$  megválasztás mellett a  $\phi(x)$  függvénynek a zérushelyei az

$$x_0, x_0, \dots, x_n, x'$$

pontokban vannak, azaz  $\phi(x)$  a 0 értéket vesz fel. A Rolle tétel kimondja, hogy ha az  $f$  függvény folytonos az  $[a, b]$  intervallumban, differenciálható az intervallum belső pontjaiban és  $f(a) = f(b)$  akkor  $\exists c$  szám, hogy  $a < c < b$  és  $f'(c) = 0$ .

A Rolle tételt hasonlóan alkalmazhatjuk  $n + 1$  darab intervallumra is. Jelen esetben ez az  $n + 1$  darab intervallum a következő:

$$[x_0, x_1], \dots, [x_i, x'], [x', x_{i+1}], \dots, [x_{n-1}, x_n]$$

ahol  $x'$  az  $x_i$  és  $x_{i+1}$  alappontok között helyezkedik el.

Ekkor  $f$  függvény deriváltja legalább  $n + 1$  helyen nulla.

Alkalmazzuk a Rolle-közéértéktételt  $\phi$  függvényre, az alappontok és  $x'$  határozzák meg az intervallumokat. Ekkor tehát  $\phi$  függvény deriváltja legalább  $n + 1$  helyen nulla, mert az  $n + 1$  alappont és  $x'$  összesen  $n + 2$  alappont, ami  $n + 1$  intervallumot határoz meg. Hasonlóan,  $\phi''$  legalább  $n$  darab pontban nulla és így tovább. Ekkor  $\phi^{(n+1)}$  legalább egy pontban nulla. Jelölje ezt az egy pontot  $\xi$ . Ekkor  $\phi^{(n+1)}(\xi) = 0$

Tehát:  $\phi^{(n+1)}(x) = f^{(n+1)}(x) - p_n^{(n+1)}(x) - (K\omega_{n+1}(x))^{(n+1)}$ . Tudjuk, hogy  $p_n^{(n+1)}(x) = 0$ , mert  $p_n$  egy  $n$ -ed fokú polinom, aminek az  $n + 1$ -edik deriváltja nulla. Tehát

$$\phi^{(n+1)}(x) = f^{(n+1)}(x) - 0 - (K\omega_{n+1}(x))^{(n+1)}.$$

Alkalmazva a szorzat függvény deriváltjára vonatkozó szabályt a következőt kapjuk:

$$\phi^{(n+1)}(x) = f^{(n+1)}(x) - (K\omega_{n+1}(x))^{(n+1)} = f^{(n+1)}(x) - K(n+1)!.$$

Tehát  $\xi$  behelyettesítve  $\phi$ -be a következőt kapjuk:

$$0 = \phi^{(n+1)}(\xi) = f^{(n+1)}(\xi) - K(n+1)!.$$

Ebből fejezzük ki  $K$ -t:

$$K = \frac{f^{(n+1)}(\xi)}{(n+1)!} = \frac{f(x') - p_n(x')}{\omega_{n+1}(x')}$$

felhasználva az előző összefüggést  $K$  meghatározására. Ezt átrendezve megkapjuk a kívánt összefüggést:

$$f(x) - p_n(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} \cdot \omega_{n+1}(x).$$

- Jelölés: Legyen

$$M_{n+1} = \max_{[a,b]} |f^{(n+1)}(x)|.$$

Ilyen  $M_n$  létezik a Weierstrass tétel szerint. Ezt felhasználva a következő összefüggést kapjuk:

$$|f(x) - p_n(x)| \leq \frac{M_{n+1}}{(n+1)!} \cdot |\omega_{n+1}(x)|, \quad n \in \mathbb{N}$$

Tovább egyszerűsíthetjük a hibabecslést, ha  $\omega_{n+1}(x)$ -re alkalmazzuk a következő becslést:

$$|\omega_{n+1}(x)| = (x - x_0) \dots (x - x_n) \leq (b - a)^{n+1}$$

Mivel  $x_i$ ,  $i = 0, \dots, n$  alappontok mind  $[a, b]$  intervallumon belül vannak ezért egy tetszőleges  $x$  intervallumon belüli pontra, a szorzat minden tagja felülről becsülhető  $(b - a)$ -val. (Megjegyzés:  $x \leq b$  és  $a \leq x_i, i = 0 \dots n$ )

Ezt felhasználva:

### 1.3.2. Tétel.

$$|f(x) - p_n(x)| \leq \frac{M_{n+1}}{(n+1)!} \cdot (b-a)^{n+1}$$

minden  $x \in [a, b]$ .

Ennél pontosabban is becsülhetjük  $\omega_{n+1}(x)$ -et: Jelölje  $h = \max_i(x_i - x_{i-1})$ -et. Ekkor

$$|\omega_{n+1}(x)| = (x - x_0) \dots (x - x_n) \leq \frac{1}{4} \cdot n! \cdot h^{n+1}$$

*Bizonyítás.* Teljes indukciós bizonyítás  $n$  szerint.

$n = 1$  esetén kettő pont van  $(x_0, x_1)$  és  $x_0 < x_1$ , ekkor

$\omega_1(x) = (x - x_0)(x - x_1)$ , tehát

$$|\omega_1(x)| \leq \frac{1}{4} \cdot 1 \cdot h^2 = \frac{1}{4} \cdot (x_1 - x_0)^2$$

és

$$|(x - x_0)(x - x_1)| \leq (x - x_0)(x_1 - x) \leq \frac{1}{4}(x_1 - x_0)^2.$$

A számtani mértani közepek közötti összefüggésből. Tehát  $n = 1$ -re beláttuk az állítást.

Tegyük fel, hogy  $n - 1$ -re teljesül az állítás, bizonyítsuk be  $n$ -re.

$$|\omega_n(x)| = |x - x_n| \cdot |\omega_{n-1}(x)| \leq |x - x_n| \cdot \frac{1}{4} \cdot (n-1)! \cdot h^n$$

felhasználva, hogy  $n - 1$ -re teljesül az összefüggés.

$(x - x_n) < n \cdot h$  -mert  $h$  a maximális távolság két pont között tehát ennek az  $n$ -szerese biztosan nagyobb, mint  $(x - x_n)$ . Ezt felhasználva:

$$|\omega_n(x)| \leq \frac{1}{4} \cdot n! \cdot h^{n+1}$$

amiből a hibára a következő összefüggést kapjuk:

$$|f(x) - p_n(x)| \leq \frac{M_{n+1}}{(n+1)!} \cdot \frac{1}{4} \cdot n! \cdot h^{n+1}.$$

Ezt egyszerűsítve:

$$|f(x) - p_n(x)| \leq \frac{M_{n+1}}{(n+1)} \cdot \frac{1}{4} \cdot h^{n+1}$$

képletet kapjuk. Ezzel beláttuk az állítást. □

**1.3.1. Definíció.** Definiáljuk a hibafüggvényt a szokásos  $e_n(x)$  jelöléssel.

$$e_n(x) = f(x) - p_n(x)$$

Vizsgáljuk, hogy  $\lim_{n \rightarrow \infty} e_n(x) = 0$  teljesül-e.

- A különböző konvergencia fajták:
  - pontonként:  $x' \in [a, b]$  rögzített pontra  $e_n(x')$  számsorozat tart-e 0-hoz.
  - egyenletesen:  $\max_{x \in [a, b]} |e_n(x)|$  számsorozat tart-e 0-hoz, ha  $n$  tart  $\infty$ -hez.

Megjegyzés: Az egyenletes konvergencia erősebb, mint a pontonkénti konvergencia, azaz az egyenletes konvergenciából következik a pontonkénti. Ezért elég belátni, hogy a hibafüggvény egyenletesen konvergens.

**1.3.3. Tétel.** Tegyük fel, hogy  $f \in C^\infty[a, b]$  és  $M_n \leq M \forall n$ -re  
akkor a  $p_n$  interpolációs polinomok sorozata egyenletesen tart  $f$ -hez  $[a, b]$ -on.

*Bizonyítás.* Egyenletes konvergencia bizonyítása:

$$|f(x) - p_n(x)| \leq \frac{M_{n+1}}{(n+1)!} (b-a)^{n+1} \leq \frac{M}{(n+1)!} (b-a)^{n+1}$$

Tudjuk, hogy  $c_n := \frac{d^n}{n!}$  jelöléssel  $\lim_{n \rightarrow \infty} c_n = 0$ . Ezt felhasználva

$$\lim_{n \rightarrow \infty} \frac{(b-a)^{n+1}}{(n+1)!} = 0,$$

tehát

$$|f(x) - p_n(x)| \leq \frac{M}{(n+1)!} (b-a)^{n+1} \rightarrow 0$$

ha  $n \rightarrow \infty$

Ebből következik, hogy

$$\max_{x \in [a, b]} |f(x) - p_n(x)| \rightarrow 0$$

ha  $n \rightarrow \infty$  tehát beláttuk az egyenletes konvergenciát, a hibafüggvény a 0-hoz tart.

□

- Megjegyzés: Ha csak annyi adott, hogy  $f \in C^\infty[a, b]$  és  $|f^n(x)| \leq c^n$   $c \in R$

$$\max_{x \in [a, b]} |f(x) - p_n(x)| \leq \frac{c^{n+1}}{(n+1)!} (b-a)^{n+1} = \frac{[c(b-a)]^{n+1}}{(n+1)!} \rightarrow 0$$

ha  $n \rightarrow \infty$

Tehát ebben az esetben is teljesül az egyenletes konvergencia.

**1.3.2. Definíció.** Azt mondjuk, hogy az interpolációs formula  $r$ -ed rendben konvergens, ha

$$\|e_n\|_\infty \leq c \cdot h^r$$

teljesül  $h = \max_{i \in [a, b]} (x_{i+1} - x_i)$  jelöléssel.

- Következmény: A Lagrange féle interpoláció  $M_n \leq M$  esetén  $n+1$ -ed rendű.

**1.3.3. Definíció.** Azt mondjuk, hogy az interpolációs formula  $q$ -ad rendben pontos, ha a  $P_q$ -beli polinomokra pontos, azaz  $e_n(x) = 0$  minden  $f \in P_q$  polinomra. Tehát a  $p_n(x) = \sum_{i=0}^n a_i x^i$ ,  $P_n \in P_q$  interpolációs polinom visszaadja az eredeti  $f$  függvényt, azaz  $p_n(x) = f(x)$  teljesül minden  $x \in [a, b]$ -re, ha  $f^{(n+1)}(\xi) = 0$

**1.3.4. Tétel.** Az  $f(x) = \sum_{i=0}^n a_i x^i \in P_n$  interpolációs formula pontos, ha  $f(x)$   $n+1$ -edik deriváltja 0, azaz  $f^{(n+1)}(\xi) = 0$ ,  $\xi \in (a, b)$ .

*Bizonyítás.* Vizsgáljuk jelen esetben a hibabecslés képletét:

$$e_n(x) = f(x) - p_n(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} \cdot \omega_{n+1}(x)$$

tagban a jobb oldal nulla a fenti összefüggés miatt, tehát a hiba is 0. Ezzel beláttuk az állítást.

□



**1.3.4. Definíció.** Azt mondjuk, hogy a  $L_n f$  Lagrange féle interpoláció konvergens az  $x^* \in [a, b]$  pontban ha igaz a következő:

$$\exists \lim_{n \rightarrow \infty} (L_n f)(x^*)$$

és egyenlő  $f(x^*)$ -al.

**1.3.5. Tétel.** *Weierstrass tétel:* Legyen  $f \in C[a, b]$  ekkor minden  $\varepsilon > 0$  esetén létezik  $p_n \in P_n$  polinom amelyre  $|f(x) - p_n(x)| < \varepsilon$  minden  $x \in [a, b]$  és  $\varepsilon = \frac{1}{n}$  választásával

$|f(x) - p_n(x)| < \frac{1}{n}$  minden  $x \in [a, b]$  amiből következik, hogy

$\|f(x) - p_n(x)\|_\infty < \frac{1}{n}$  amiből következik, hogy ha  $n \rightarrow \infty$ ,

akkor  $\|f(x) - p_n(x)\|_\infty \rightarrow 0$

**1.3.6. Tétel.** *Fabre tétel:* Legyen  $f \in C[a, b]$ . Ekkor az  $[a, b]$  intervallum bármely  $\Omega_n$  felosztásához  $\exists f \in C[a, b]$  amire az  $L_n f$  Lagrange féle interpolációs sorozat nem tart egyenletesen  $f$ -hez.  $\Omega_n := \{a = x_0 < \dots < x_n = b\}$  egy felosztása  $[a, b]$  intervallumnak.

**1.3.7. Tétel.** *Marczinkievicz tétel:* minden  $f \in C[a, b]$  létezik  $[a, b]$  intervallum egy  $\Omega_n$  felosztása amelyre az  $L_n f$  Lagrange féle interpolációs sorozat tart egyenletesen  $f$ -hez.

- Hogyan tudnánk csökkenteni a hibát?
  - Hogyan válasszuk meg  $x_i, i = 0, \dots, n$  alappontokat a pontosság növelése érdekében?
  - Mi történik, ha nem rögzített, hanem változó alappontokon vesszük az interpolációt? (Csebisev alappontokon történő interpoláció)
  - Hogyan befolyásolja a hibát, ha rögzített alappontokon nézzük az interpolációt, de a függvényről több információval rendelkezünk. (például a deriváltjainak értékeivel az alappontokban). (Hermite-féle interpoláció)

## 1.4. Interpoláció Csebisev-féle alappontokon

- Vizsgáljuk, hogyan befolyásolja az interpolációs hibát, az alappontok speciális megválasztása. Nézzük az interpolációt Csebisev alappontokon.
- Csebisev alappontok: Legyen  $p_n(x) = \cos(n \cdot \arccos(x))$ ,  $x \in [-1, 1]$ -re. Megjegyzés: A későbbiekben kiterjeszthetjük tetszőleges  $[a, b]$  intervallumra.

**1.4.1. Tétel.** *Állítás  $p_n$ -re teljesül a következő rekurzió:*

$$p_n(x) = 2x \cdot p_{n-1}(x) - p_{n-2}(x)$$

*Bizonyítás.* Legyen  $\alpha = \arccos(x)$  Ekkor:

$$p_{n+1} = \cos(n+1)\alpha = \cos(n\alpha + \alpha) = \cos(n\alpha)\cos(\alpha) - \sin(n\alpha)\sin(\alpha)$$

alkalmazva az addíciós tételket a szögfüggvényekre.

Hasonlóan:

$$p_{n-1} = \cos(n-1)\alpha = \cos(n\alpha - \alpha) = \cos(n\alpha)\cos(\alpha) + \sin(n\alpha)\sin(\alpha)$$

Ezek összege:

$$\cos(n+1)\alpha + \cos(n-1)\alpha =$$

$$\cos(n+1)\alpha + \cos(n-1)\alpha =$$

$$\cos(n\alpha)\cos(\alpha) - \sin(n\alpha)\sin(\alpha) + \cos(n\alpha)\cos(\alpha) + \sin(n\alpha)\sin(\alpha) =$$

$$\begin{aligned}\cos(n\alpha)\cos(\alpha) + \sin(n\alpha)\sin(\alpha) &= \\ &= 2\cos(n\alpha)\cos(\alpha)\end{aligned}$$

Tehát felhasználva  $p_n$  definícióját és a fenti összefüggéseket a következőt kapjuk:

$$p_{n+1}(x) + p_{n-1}(x) = 2p_n(x)\cos(\arccos(x)) = 2x \cdot p_n(x)$$

mivel  $\cos(\arccos(x)) = x$

Ezt átalakítva:

$$p_{n+1}(x) = 2x \cdot p_n(x) - p_{n-1}(x)$$

Továbbá:

$$p_0(x) = \cos(0 \cdot \arccos(x)) = \cos(0) = 1$$

$$p_1(x) = \cos(1 \cdot \arccos(x)) = x$$

és így tovább:

$$p_2(x) = 2x \cdot p_1(x) - p_0(x) = 2x^2 - 1$$

Tehát beláttuk, hogy teljesül a rekurzió. □

Hasonlóan folytatva az eljárást láthatjuk, hogy  $p_n(x)$  valóban  $n$ -ed fokú.

**1.4.2. Tétel.** *Mutasuk meg, hogy  $p_n \in \mathbb{P}_n$*

*Bizonyítás.* A bizonyítás típusa teljes indukciós bizonyítás  $n$ -re.  $n = 0$ -ra és  $n = 1$ -re teljesül ezt a fenti számolásból láthatjuk. Tegyük fel, hogy  $n - 1$ -ig igaz, bizonyítsuk be, hogy  $n$ -re is igaz.

$$p_n(x) = 2x \cdot p_{n-1}(x) - p_{n-2}(x)$$

Mivel  $p_{n-1}$   $n - 1$ -ed fokú, ezért  $x \cdot p_{n-1}$   $n$ -ed fokú lesz, mivel  $p_{n-2}$  csak  $n - 2$ -ed fokú ezért ezzel beláttuk az állítást. □

- Vegyük észre továbbá, hogy  $p_n(x)$  olyan  $n$ -ed fokú polinom, amelyben  $x^n$  együtthatója  $2^{n-1}$ . Ezt az állítást könnyű bizonyítani teljes indukcióval az előzőhöz hasonló módon.

**1.4.1. Definíció.** A

$$T_n(x) = 2^{1-n} \cos(c \cdot \arccos(x)) \in P_n$$

polinomot a Csebisev féle polinomnak nevezzük.

- Vizsgáljuk ezt a Csebisev féle polinomot. A  $T_n(x)$  zérushelyei az alábbi egyenlet  $x$  megoldásai:

$$2^{1-n} \cos(c \cdot \arccos(x)) = 0.$$

Ez a kifejezés pontosan akkor 0, ha

$$n \cdot \arccos(x) = \frac{2k+1}{2} \pi$$

azaz  $\arccos(x) = \frac{2k+1}{2n} \pi$   $k \in \mathbb{Z}$

Tehát

$$x_k = \cos\left(\frac{2k+1}{2n} \pi\right), \quad k = 0, \dots, n-1$$

**1.4.3. Tétel.** Legyen  $Q_n \in \mathbb{P}_n$  olyan tetszőleges polinom, amelynek  $x^n$  együtthatója 1, ekkor

$$\|T_n\|_\infty \leq \|Q_n(x)\|_\infty$$

*Bizonyítás.* Tehát bizonyítsuk be, hogy az 1 főegyütthatós polinomok közül a  $T_n$ -nek a legkisebb a maximum normája.  $T_n(x) = (x - x_0) \dots (x - x_{n-1})$  azonban  $\|T_n(x)\|_\infty = 2^{1-n}$  mivel  $\cos$  függvény maximális értéke 1. Ennek alapján a már definiált  $\omega_n$  függvényt becsülhetjük a következő módon:

$\|\omega_{n+1}\|_\infty$  akkor a legkisebb ha  $\omega_{n+1} = T_{n+1}(x)$ . Ekkor

$$\|\omega_{n+1}\|_\infty = 2^{1-(n+1)} = 2^{-n}.$$

Mivel a Csebisev polinom gyökei az alappontok ebből következnek, hogy ha  $f \in C^{n+1}(a, b)$  interpolációs becsléséhez az  $x_k = \cos(\frac{2k+1}{2n}\pi)$ ,  $k = 0 \dots n$  alappontokat választjuk, akkor igaz, hogy

$$\|f - L_N f\|_\infty \leq \frac{M_{n+1}}{(n+1)!} 2^{-n}.$$

□

Megjegyzés: Eddig feltettük, hogy  $f$  abszolút folytonos, ekkor lesz igaz a fenti állítás. Azonban ha  $f$ -re nem tesszük fel, hogy legyen abszolút folytonos, akkor az interpoláció a Csebisev-alappontokon sem konvergens, a Fabre-tétel miatt, nem igaz, hogy minden  $f \in [a, b]$  esetén  $L_n f$  egyenletesen tart  $f$ -hez.

**1.4.2. Definíció.** Az  $f$  függvény abszolút folytonos az  $I \subset \mathbb{R}$  intervallumon minden  $\varepsilon$ -hoz létezik  $\sigma > 0$  amelyre teljesül, hogy minden olyan  $x_i, x_j \in I$ -re amelyre  $|x_i - x_j| < \sigma$  teljesül, hogy  $|f(x_i) - f(x_j)| < \varepsilon$

## 1.5. Hermite interpoláció

Az eddigiekben azt vizsgáltuk, hogy ha ismert egy függvény  $n+1$  alappontban, akkor hogyan határozhatjuk meg, azt a polinomot, ami áthalad az adott alappontokon. A célunk továbbra is, az, hogy a hibát csökkentsük. Vizsgáljuk meg, hogy ha az alappontokban nem csak a függvényértékeket, hanem a pontokban vett deriváltakat is ismerjük egy adott rendig bezárólag, akkor hogyan tudjuk az eredeti függvényt közelíteni. Azaz az  $x_0, \dots, x_n$  alappontokhoz legyen adott  $m_k + 1$  darab érték:

$f_k^{(0)}, f_k^{(1)}, \dots, f_k^{(m_k)}$ . Ekkor keressük azt a  $H$  polinomot, amelyre

$$H^{(i)}(x_k) = f_k^{(i)}, \quad k = 0 \dots n, \quad i = 0 \dots m_k$$

Ezt a módszert nevezzük Hermite-féle interpolációnak. Megjegyezzük, hogy  $m_k = 0$  esetén polinom interpolációról beszélünk, mert ekkor a feladat a következő:

$$H^{(1)}(x_k) = f_k^{(1)} = f_k, \quad k = 0 \dots n, \quad i = 0$$

és a nulladik derivált éppen a függvényérték. Ha minden alappontban a függvényértékek (0. derivált) és az első deriváltak adottak, akkor a Hermite-Fejér interpolációról beszélünk. Mivel összesen  $N = m_0 + \dots + m_n + n + 1$  adott adat van, ezért várhatóan  $(N - 1)$ -ed fokú lesz a megfelelő  $H$  polinom.

**1.5.1. Tétel.** Egyértelműen létezik  $H_{N-1} \in \mathbb{P}_{N-1}$  polinom, amelyre teljesül az adott feltétel:

$$H_{N-1}^{(i)}(x_k) = f_k^{(i)}, \quad k = 0 \dots n, \quad i = 0 \dots m_k$$

*Bizonyítás.* Legyen  $H_{N-1}(x) = \sum_{i=0}^{N-1} a_i x^i$  Ekkor az  $a_i$  együtthatók meghatározásához a következő egyenletrendszert kell megoldani:

$$\begin{bmatrix} 1 & x_0 & x_0^2 & \dots & x_0^{N-1} \\ 0 & 1 & 2x_0 & \dots & (N-1)x_0^{N-2} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ 1 & x_1 & x_1^2 & \dots & x_1^{N-1} \\ 0 & 1 & 2x_1 & \dots & (N-1)x_1^{N-2} \\ \vdots & \vdots & \vdots & \dots & \vdots \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_{N-1} \end{bmatrix} = \begin{bmatrix} f_0^{(0)} \\ f_0^{(1)} \\ \vdots \\ f_1^{(0)} \\ f_1^{(1)} \\ \vdots \\ \vdots \end{bmatrix}$$

A fenti egyenletben legyen:  $M \in \mathbb{R}^{N \times N^2}$  mátrix,  $a \in \mathbb{R}^N$  vektor, míg  $f \in \mathbb{R}^{N^2}$  vektor.

Ez összesen  $N$  darab egyenlet,  $N$  darab ismeretlenre és az együttható mátrix nem szinguláris (invertálható), mert ha létezne olyan  $a \neq 0$  vektor amelyre  $M \cdot a = 0$  akkor  $H_{N-1}$ -nek  $N$  darab zérushelye lenne ami ellentmond az algebra alaptételének.

Megjegyzés: Ha egy polinomnak  $x$   $k$ -szoros gyöke, akkor  $x$  a polinom első  $k-1$  deriváltjának is zérushelye.  $\square$

- A Hermite–Fejér interpoláció:

A Hermite–Fejér interpolációnál minden  $x_0, \dots, x_n$  alappontban csak a függvényértékek (0. deriváltak) illetve az első deriváltak értékei adottak. Ez összesen  $2n+2$  adat. Tehát egy legfeljebb  $2n+1$ -ed fokú polinom lesz a megfelelő interpolációs polinom. Jelöljük a keresett polinomot  $H_{2n+1}(x)$ -el.

- Megjegyzés: A Hermite–Fejér interpolációs polinomot megkaphatjuk úgy, hogy megoldjuk az előző fejezetben található egyenletrendszert, de ez a módszer nem hatékony, mert magas a műveletigénye. Keressünk kisebb műveletigényű módszert:

**1.5.2. Tétel.** *A Hermite-Fejér interpolációs polinom el áll a következő alakban:*

$$H_{2n+1}(x) = \sum_{i=0}^n f_i^{(0)} (1 - 2(x - x_i)l_i'(x_i))l_i^2(x) + \sum_{i=0}^n f_i^{(1)} (x - x_i)\ell_i^2(x)$$

ahol a  $\ell_i(x)$  az  $x_i$  alapponthoz tartozó Lagrange-féle alappolinom.

*Bizonyítás.* Vegyük azokat a  $P_{2n+1}$ -beli polinomokat, amelyek valamelyik alappontban az értékük, vagy a deriváltjának értéke 1 míg az összes többi alappontban 0. Jelölje  $h_{k0}$  azt a polinomot, amelyre

$$h_{k0}(x_i) = \delta_{ki}, \quad h'_{k0}(x_i) = 0$$

továbbá  $h_{k1}$  azt a polinomot, amelyre:

$$h'_{k1}(x_i) = \delta_{ki}, \quad h_{k1}(x_i) = 0$$

ahol  $k, i = 0 \dots n$  és  $\delta_{ki}$  a szokásos Kronecker-delta, azaz értéke 1 ha  $k = i$ , és értéke 0 ha  $k \neq i$ .

Keressük az alappolinomokat  $h(x) = s(x)\ell_i^2(x)$  alakban ahol  $\ell_i(x)$  az  $i$ . alapponthoz tartozó Lagrange-féle alappolinom. Ekkor  $s(x)$  az ismeretlen első fokú polinom. Deriváljuk  $h(x)$ -et, ekkor a szorzat deriválási szabálya miatt a következőt kapjuk:

$$h'(x) = s'(x)\ell_i^2(x) + 2s(x)\ell_i(x)\ell_i'(x)$$

Legyen  $h(x)$  olyan, hogy az  $x_i$ -től különböző alappontban az értéke és a deriváltja legyen 0, míg az  $x_i$  alappontban az értéke 0, a deriváltja pedig 1. Ekkor tehát az  $s(x) = x - x_i$  a megfelelő választás, mert  $s(x_i) = 0$  és  $s'(x_i) = 1$ .

Ekkor tehát

$$h_{i1}(x_i) = (x - x_i)\ell_i^2(x), \quad i = 0, \dots, n.$$

Ha  $h(x)$ -nek azt a feltételt is teljesítenie kell, hogy  $x_i$ -től különböző alappontban az értéke és a deriváltja is 0, továbbá  $x_i$ -ben az értéke 1 a deriváltja pedig 0 legyen, akkor a következőnek kell teljesülnie:

$s(x_i) = 1$  és  $s'(x_i) = -2\ell_i'(x_i)$ . Tehát  $s(x) = 1 - 2(x - x_i)\ell_i'(x_i)$  jó választás lesz. Tehát

$$h_{i0}(x) = (1 - 2(x - x_i)\ell_i'(x_i))\ell_i^2(x).$$

Ekkor tehát az interpolációs polinom a következő lesz:

$$H_{2n+1}(x) = \sum_{i=0}^n f_i^{(0)} h_{i0}(x) + \sum_{i=0}^n f_i^{(1)} h_{i1}(x)$$

Ezt akartuk belátni. □

- A Hermite–Fejér interpolációs módszer hibája:

**1.5.3. Tétel.** *Tegyük fel, hogy  $f \in C^{2n+2}([a, b])$ . Ekkor az  $a = x_0, \dots, x_n = b$  alappontokon interpoláljuk az  $f$  függvényt a Hermite–Fejér interpolációval, akkor ennek hibája:*

$$E_n(x) = H_{2n+1}(x) - f(x) = -\frac{f^{(2n+2)}(\xi_x)}{(2n+2)!} \omega_{n+1}^2(x),$$

ahol  $\xi_x \in [a, b]$  konstans, amelynek értéke függ  $x$  megválasztásától.

## 1.6. Szakaszonként polinomiális interpoláció (Spline interpoláció)

Az előző fejezetben azzal is foglalkoztunk, hogyan tudjuk az interpoláció hibáját csökkenteni, az alappontok speciális megválasztásával. Azonban ha az alappontok adottak, mert például egy mérés eredményeként kapjuk őket, akkor nem alkalmazhatjuk a Csebisev-alappontokat. Ha az alappontjaink mérések eredményei, akkor a Hermite interpolációt sem alkalmazhatjuk, mert ott hiába rögzített alappontokon történik az interpoláció, a deriváltakra vonatkozó többlet információ nem áll rendelkezésre. Továbbá a dolgozat utolsó fejezetében tárgyalt feladatok megoldásánál is láthatjuk, hogy az interpolációs polinom sokszor nem olyan alakú, nem úgy köti össze a pontokat, mint ahogy szeretnénk, vagy mint amilyen az eredeti függvény. (Túlzott hullámozása az interpolációs polinomnak). Például ha egymáshoz közel eső pontokhoz azonos  $f$  érték tartozik, akkor a polinom ezen a szakaszon nagyon eltérhet az eredeti értékektől. Továbbá, ha a pontok monoton növekvő  $f$  értékekkel rendelkeznek, abból nem feltétlen következik, hogy az interpolációs polinom is monoton növekvő lesz. Ezen hibák kiküszöbölésére alkalmazzuk a Szakaszonként polinomiális interpolációt. Ekkor a szomszédos alappontok közötti szakaszokon egy alacsony fokú polinommal interpoláljunk. Ezt a módszert nevezzük Spline interpolációnak.

- Mivel a polinomiális interpolációnál (Lagrange vagy Newton)  $n + 1$  ponthoz egy  $n$ -ed fokú polinom tartozik ( $L_n = N_n$ ) továbbá ennek a polinomnak át kell haladnia az  $x_0, \dots, x_n$  alappontokon. Nagy  $n$  értékre az interpolációs polinom magas fokú ami további pontatlanságot eredményezhet. Ahhoz, hogy áthaladjon az alappontokon, a szomszédos pontok között a polinom sokszor nem megfelelően sima, azaz a polinom nagyon hullámzik a szomszédos pontok között. Ennek a kiküszöbölésére is alkalmazhatjuk a Spline interpolációt. Tehát megköveteljük, hogy az alappontok között az interpolációs polinom kellő simaságú legyen.

- Szakaszontként lineáris interpoláció:

Ez a legegyszerűbb Spline-interpolációs technika. Ekkor tehát a szomszédos pontokat egy-egy elsőfokú polinommal azaz egy egyenes szakasszal kötjük össze. Legyenek az alappontok (sorba rendezve)  $a = x_0 < \dots < x_n = b$ , továbbá hasonlóan az eddig jelöléshez a pontokban a függvényértékeket jelöljük:  $f_0, \dots, f_n$ -el. Továbbá az alappontok által meghatározott intervallumokon  $[x_{i-1}, x_i], i = 1, \dots, n$  az interpolációs Spline polinom legyen  $s_i, i = 1, \dots, n$ . Ekkor ha szakaszontként lineáris interpolációt alkalmazunk, akkor nyilvánvalóan:

$$s_i(x) = f_{i-1} \frac{x - x_i}{x_{i-1} - x_i} + f_i \frac{x - x_{i-1}}{x_i - x_{i-1}}, \quad x \in [x_{i-1}, x_i]$$

Ekkor tehát a teljes  $[x_0, x_n]$  intervallumra értelmezett interpolációs függvény a következő alakban adható meg:

$$s(x) = \begin{cases} s_1(x), & \text{ha } x \in [x_0, x_1] \\ s_2(x), & \text{ha } x \in [x_1, x_2] \\ \vdots \\ s_n(x), & \text{ha } x \in [x_{n-1}, x_n] \end{cases}$$

Az  $s$  folytonos függvény, mert

$$s_i(x_i) = s_{i+1}(x_i) = f_i$$

Azaz az alappontokban a bal illetve jobb oldali deriváltak megegyeznek.

- A lineáris Spline interpoláció hibája:

Legyen  $f \in C^2([a, b])$  Ekkor a lineáris Spline interpolációs függvény hibája

$$|s(x) - f(x)| \leq \frac{M_2}{8} h^2$$

ahol  $M_2 = \max_{y \in [a, b]} f''(y)$ , azaz  $M_2 \geq f''(y)$  minden  $y \in [a, b]$ -re ( $f$  második deriváltjának maximuma  $[a, b]$  intervallumon), és  $h$  pedig a szomszédos alappontok közötti távolságok maximuma, azaz  $h = \max_{i=1..n} (x_i - x_{i-1})$

- Megjegyzés: A módszer hátránya, hogy a kapott függvény nem deriválható az intervallumon. Ennek kiküszöbölésére alkalmazhatjuk a Kvadratikus Spline interpolációt
- A szakaszontként kvadratikus Spline interpoláció: A szomszédos alappontokat nem első, hanem másodfokú polinommal kötjük össze. Adjuk meg az interpolációs függvény deriváltjának értékét az  $x_0 = a$  kezdőpontban. Legyen ez az érték  $d_0$ . Ekkor alkalmazzuk a Hermite-interpolációt  $[x, x_1]$  intervallumra, ez egyértelműen meghatároz egy  $s_1(x)$  legfeljebb másodfokú polinomot, amelyre teljesül, hogy az  $x_0$  és  $x_1$  alappontokban az értéke rendre  $f_0$  és  $f_1$  (interpolációs alapfeltételt teljesíti). Továbbá  $s_1'(x) = d_0$ . Mivel  $s_1$  adott és deriválható  $[x_0, x_1]$ -en, ezért létezik  $s_1'(x_1)$ , nevezzük el  $d_1$ -nek ezt az értéket. Ezután alkalmazzuk a Hermite-interpolációt ezután az  $[x_1, x_2]$  intervallumra, így hasonlóan mint az előbb egy legfeljebb másodfokú polinomot ( $s_2(x)$ ) kapunk az  $[x_1, x_2]$  intervallumon, ami teljesíti az interpolációs alapfeltételt. Mivel  $s_2$  szintén differenciálható az intervallum minden belső pontjában, ezért létezik  $s_2'(x_2)$ , legyen ez  $d_2$ . A többi intervallumra folytatva az eljárást, és az előző módon konstruált  $s$  függvény deriválható, az egész  $[a, b]$  intervallumon, és deriváltja folytonos.
- A Szakaszontként harmadfokú interpoláció: Ebben az esetben minden intervallumon egy legfeljebb harmadfokú polinomot alkalmazunk az interpolációra. Ekkor, ha megadjuk  $x_0 \dots x_n$  pontokhoz tartozó  $d_0 \dots d_n$  értékeket, akkor tehát minden intervallum kezdő és végpontjában adott egy függvényérték és egy deriváltérték. Ezek segítségével egyértelműen meghatározhatunk egy harmadfokú polinomot, a Hermite-Fejér interpoláció segítségével.

Ezzel az eljárással folytonosan deriválható interpolációs függvényt kapunk. A  $d_0, \dots, d_n$  számhoz megfelelő megválasztásával továbbá elérhető az is, hogy a második deriváltak is folytonosak legyenek. Legyen  $s_i$  az  $[x_{i-1}, x_i]$  intervallumon adott legfeljebb harmadfokú polinom  $i = 1, \dots, n$  esetén. Ekkor az adott  $d_{i-1}, d_k$  deriváltértékekkel az  $s_k$  polinomot a következő képpen adhatjuk meg:

- Hermite–Fejér interpoláció:

alappontok	$f_i = [\cdot]f$	$[\cdot, \cdot]f$	$[\cdot, \cdot, \cdot]f$	$[\cdot, \cdot, \cdot, \cdot]f$
$x_{i-1}$	$f_{i-1} = c_{i0}$	$d_{i-1} = c_{i1}$		
$x_{i-1}$	$f_{i-1}$		$\frac{f_i - f_{i-1} - d_{i-1}}{x_i - x_{i-1}} = c_{i2}$	
		$\frac{f_i - f_{i-1}}{x_i - x_{i-1}}$		$\frac{d_i - \frac{f_i - f_{i-1}}{x_i - x_{i-1}} - \frac{f_i - f_{i-1} - d_{i-1}}{x_i - x_{i-1}}}{x_i - x_{i-1}} = c_{i3}$
$x_i$	$f_i$		$d_i - \frac{f_i - f_{i-1}}{x_i - x_{i-1}}$	
$x_i$	$f_i$	$d_i$		

Ezeket a jelöléseket használva:

$$s_k(x) = c_{i0} + c_{i1}(x - x_{i-1}) + c_{i2}(x - x_{i-1})^2 + c_{i3}(x - x_{i-1})^2(x - x_i)$$

alakban írható fel az  $[x_{i-1}, x_i]$  intervallumon. Hasonló képpen felírhatjuk  $s_{i+1}$  polinomot is:

$$s_{i+1}(x) = c_{i+1,0} + c_{i+1,1}(x - x_{i-1}) + c_{i+1,2}(x - x_{i-1})^2 + c_{i+1,3}(x - x_{i-1})^2(x - x_i).$$

Ezeket a kifejezéseket deriválva kapjuk, hogy:

$$s_i''(x) = 2c_{i2} + 2c_{i3}(x - x_i) + 4c_{i3}(x - x_{i-1})$$

és

$$s_{i+1}''(x) = 2c_{i+1,2} + 2c_{i+1,3}(x - x_i) + 4c_{i+1,3}(x - x_{i-1}).$$

A következőkben a második deriváltak folytonosságának feltételét vizsgáljuk meg. Az  $x_i$  alappontban a következőnek kell teljesülnie: Mivel

$$s_i''(x_i) = s_{i+1}''(x_i),$$

ezért

$$2c_{i2} + 4c_{i3}(x_i - x_{i-1}) = 2c_{i+1,2} + 2c_{i+1,3}(x_i - x_{i+1}).$$

Helyettesítsük be az együtthatókat:

$$\begin{aligned} & \frac{2}{x_i - x_{i-1}} d_{i-1} + 4 \cdot \left( \frac{1}{x_i - x_{i-1}} + \frac{1}{x_{i+1} - x_i} \right) d_i + \frac{2}{x_{i+1} - x_i} d_{i+1} = \\ & = 6 \left( \frac{f_i - f_{i-1}}{(x_i - x_{i-1})^2} + \frac{f_{i+1} - f_i}{(x_{i+1} - x_i)^2} \right). \end{aligned}$$

Tehát  $i = 1 \dots n-1$  esetén a fenti összefüggés egy egyenlet rendszer. Tehát amennyiben  $d_i, i = 0 \dots n$  teljesíti a fenti feltételt akkor a második deriváltak folytonosak lesznek az intervallumon.

Összesen tehát  $n - 1$  egyenletünk van és  $n + 1$  ismeretlenünk (a derivált értékek). Tehát még két feltételt meg kell adni az egyértelműséghez. Adjuk meg a kezdeti illetve a végpontokban a derivált értékeket. Legyen tehát a kezdeti pontban a második derivált érték  $D_0$  és a végpontban  $D_n$ . (Az egyértelműséget máshogyan is biztosíthatjuk, például a kezdő és végpontokban a derivált értéket adjuk meg)

Tehát  $k = 1$  esetén

$$s_1''(x_0) = 2c_{12} + 2c_{13}(x_0 - x_1) = D_0$$

az együtthatókat behelyettesítve pedig:

$$s_1''(x_0) = \frac{2}{x_1 - x_0}d_0 + \frac{1}{x_1 - x_0}d_1 = 3\frac{f_1 - f_0}{(x_1 - x_0)^2} - \frac{D_0}{2}$$

az alábbi összefüggést kapjuk. Továbbá  $k = n$  esetén hasonlóan:

$$s_n''(x_n) = 2c_{n2} + 4c_{n3}(x_n - x_{n-1}) = D_n$$

és

$$s_n''(x_n) = \frac{1}{x_n - x_{n-1}}d_{n-1} + \frac{2}{x_n - x_{n-1}}d_n = 3\frac{f_n - f_{n-1}}{(x_n - x_{n-1})^2} + \frac{D_n}{2}.$$

Tehát a fentiekkel kiegészítve már megoldható a 3.0 egyenletrendszer. A megoldás pedig olyan  $d_0 \dots d_n$  értékeket ad, amelyekre a második deriváltak is folytonosak lesznek, illetve a végpontokban a derivált értékek rendre  $D_0$  és  $D_n$ .



## 2. fejezet

# A közönséges differenciálegyenletek kezdetiérték feladatainak numerikus megoldása

A differenciálegyenleteket gyakran a természettudományos, műszaki, gazdasági folyamatok leírására használjuk. Az ilyen modellek elemzésével a közönséges differenciálegyenletek, illetve a parciális differenciálegyenletek elmélete foglalkozik. Ezek a vizsgálatok főleg a fent említett feladatok megoldhatóságával foglalkoznak. Azaz azt vizsgáljuk, hogy az adott feladat milyen feltételek mellett korrekt kitűzésű. A megoldás konkrét előállítására zárt alakban (azaz megoldása olyan képletek segítségével, amelyek könnyen kiértékelhető függvényeket tartalmaznak) csak ritkán lehetséges. Azonban ezen differenciálegyenleteket is szeretnénk vizsgálni, ezért a megoldásukat numerikus módszerrel közelíthetjük. A következő részben olyan módszerekkel foglalkozunk, amelyek a numerikus megoldást nagy pontossággal előállítják. Tehát az elemi módszerekkel nem megoldható differenciálegyenletek pontos megoldása, illetve a numerikus közelítéssel kapott megoldás eltérését is vizsgálhatjuk.

### 2.1. A közönséges differenciál egyenletek kezdetiértékfeladata

Az adott folyamatot leíró differenciálegyenlet értékét egy kezdeti időpontban ismertnek tekintjük. Ezt az időpillanatot általában 0-nak választjuk, jele  $u(0)$ . Így az adott kezdeti érték segítségével felírhatjuk a közönséges differenciálegyenlet kezdetiérték feladatának megoldását.

**2.1.1. Definíció.** Legyen  $G \subset \mathbb{R} \times \mathbb{R}^d$  egy tartomány, azaz egy nyílt összefüggő halmaz. Ekkor  $(t_0, u_0^*) \in G$  egy adott pont ( $t_0 \in \mathbb{R}, u_0^* \in \mathbb{R}^d$ ) és  $u(t_0) = u_0^*$  továbbá  $f : G \rightarrow \mathbb{R}^d$  egy folytonos leképezés. A

$$\frac{du(t)}{dt} = f(t, u), \quad u(t_0) = u_0^* \in \mathbb{R}^d$$

feladatot kezdetiérték feladatnak, vagy Cauchy feladatnak nevezzük.

- Tehát  $u_0^*$  egy  $d$  elemű vektor.

Egy Cauchy feladat megoldásakor meghatározzuk az összes olyan  $u : \mathbb{R} \rightarrow \mathbb{R}^d$  függvényt, amely valamely  $I \subset \mathbb{R}$  intervallum pontjaiban behelyettesíthető a fenti feladatba, illetve ki is elégíti azt.

**2.1.2. Definíció.** Legyen  $u : I \rightarrow \mathbb{R}^d$ , folytonosan differenciálható függvény továbbá  $I \subset \mathbb{R}$  egy nyílt intervallum. Ha  $u$ -ra teljesülnek a következő állítások:

- $\{(t, u(t)) : t \in I\} \subset G$ ;
- $\frac{du(t)}{dt} = f(t, u(t))$ , minden  $t \in I$
- $t_0 \in I$  és  $u(t_0) = u_0^*$

akkor  $u$ -t a Cauchy feladat megoldásának nevezzük.

- Amikor a Cauchy feladat egy természettudományos, műszaki vagy közgazdasági folyamat matematikai modellje, akkor alapvető elvárás, hogy a feladatnak legyen egyértelmű megoldása. Ennek biztosítására vezessük be  $\alpha$ ,  $\beta$  pozitív számok mellett a

$$H_{\alpha, \beta}(t_0, u_0^*) = \{(t, u) : |t - t_0| \leq \alpha \|u - u_0^*\|_\infty \leq \beta\} \subset G$$

jelölést. Tehát  $H_{\alpha, \beta}(t_0, u_0^*)$  egy  $(t_0, u_0^*)$  középi zárt  $d + 1$  dimenziós téglá. Mivel  $f$  folytonos az így definiált zárt  $H_{\alpha, \beta}(t_0, u_0^*)$  halmazon, ezért létezik  $f(t, u)$ -nak maximuma a halmazon. Legyen ez a maximum

$$M = \max_{H_{\alpha, \beta}(t_0, u_0^*)} \|f(t, u)\|_\infty$$

Ekkor minden  $t, |t - t_0| \leq \min\{\alpha, \frac{\beta}{M}\}$  esetén létezik a Cauchy feladatnak  $u(t)$  megoldása. Ha emellett a  $H_{\alpha, \beta}(t_0, u_0^*)$  halmazon az  $f$  függvény második változójában Lipschitzes, azaz valamely  $L > 0$  állandóra minden  $(t, u_1), (t, u_2) \in H_{\alpha, \beta}(t_0, u_0^*)$  pontban teljesül, hogy:

$$\|f(t, u_1) - f(t, u_2)\|_\infty \leq L \|u_1^* - u_2^*\|_\infty.$$

Ha ez a Lipschitz-féle feltétel teljesül, akkor a megoldás egyértelmű is.

- A továbbiakban mindig feltesszük, hogy a fenti feltételek teljesülnek, azaz mindig létezik  $H_{\alpha, \beta}(t_0, u_0^*) \subset G$  részhalmaz, amelyen  $f$  folytonos és második változójában Lipschitz, ekkor tehát egyértelműen létezik megoldás. Legyen  $I_0 := \{t \in I : |t - t_0| \leq T\}$  halmaz, ahol  $T = \min\{\alpha, \frac{\beta}{M}\}$ . A közelítő megoldást ezen az  $I_0$  intervallumon állítjuk elő.
- Általában a  $t$  változó időt jelöl, ezért a Cauchy feladat megoldása azt adja meg, hogy a rendszer hogyan változik az időben. Tehát ismerve a rendszer állapotát egy adott  $t_0$  időpillanatban, megadhatjuk azt egy  $t_0$ -t követő  $t$  időpontban is. Tehát, ha  $u(t_0)$  ismert, akkor a feladat az, hogy meghatározzuk  $u(t)$ -t, ahol  $t > t_0$ . A  $t = t_0$  időpontot kezdőpontnak, míg a megoldásfüggvény  $t_0$ -beli értékét kezdeti értéknek nevezzük. Továbbá a  $(t_0, u(t_0))$  párt kezdeti feltételnek nevezzük. Legyen  $t_0 = 0$ , így a feladat megoldásának értelmezési tartománya a  $[0, T] \subset I$  intervallum. Ekkor a feladat a következő:

$$\frac{du(t)}{dt} = f(t, u(t)), t \in [0, T], u(0) = u_0^* \quad (2.1)$$

Célunk, hogy meghatározzuk  $u(t)$  megoldásfüggvényt.

- Megjegyzés: Ha az  $f$  függvény folytonos, akkor a Cauchy feladat  $u(t)$  megoldása egyszerűen folytonosan differenciálható, azaz  $u \in C^1[0, T]$ . Ha  $f$   $p$ -ed rendben sima, akkor  $u \in C^{p+1}[0, T]$ , ahol  $p \in \mathbb{N}$ . Így az adott  $f$  függvény megfelelő simaságával biztosítható a megoldás megfelelő simasága is. Tehát feltehetjük mostantól, hogy a megoldás megfelelően sima.
- A könnyebb áttekinthetőség kedvéért mostantól legyen  $d = 1$ , azaz skaláris esetet nézzük. Ekkor a feladat átfogalmazható a következőre:  
Legyen  $Q_T := [0, T] \times \mathbb{R} \subset \mathbb{R}^2, f : Q_T \rightarrow \mathbb{R}$ . A továbbiakban a

$$\frac{du}{dt} = f(t, u), u(0) = u_0 \quad (2.2)$$

feladatot nevezzük Cauchy feladatnak, és feltesszük, hogy  $f \in C(Q_T)$  és második változójában Lipschitzes.

Második változóban Lipschitzesség a következőt jelenti:

$$|f(t, u_1) - f(t, u_2)| \leq L|u_1 - u_2|, \forall (t, u_1), (t, u_2) \in Q_T$$

továbbá  $u_0 \in \mathbb{R}$  adott szám. Tehát a feladat a következő, keressük azt a megfelelően sima  $u : [0, T] \rightarrow \mathbb{R}$  függvényt, amelyre:

$$\frac{du(t)}{dt} = f(t, u(t)), \forall t \in [0, T], u(0) = u_0.$$

- Hogyan biztosítható, hogy az  $f$  függvény második változójában Lipschitzes legyen? Tegyük fel, hogy  $f : \mathbb{R}^2 \rightarrow \mathbb{R}$  függvény az értelmezési tartománynak valamely nyílt  $H_f$  részhalmazán a második változójában korlátos deriválttal rendelkezik. Ekkor a Lagrange-közéértéktétel értelmében valamely  $y^* \in (y_1, y_2)$  érték mellett  $f(x, y_1) - f(x, y_2) = \partial_2 f(x, y^*)(y_1 - y_2)$ , azaz a Lipschitzesség teljesül a második változóban,

$$L = \sup_{H_f} (|\partial_2 f(x, y)|) < \infty$$

állandóval.

Tehát ekkor a Cauchy-feladatnak  $f$  létezik egyértelmű megoldása a  $[0, T]$  intervallumon.

## 2.2. Egylépéses módszerek

Az előző részben a megoldás létezésére és egyértelműségére való feltételeket vizsgáltuk, azonban ezek az előállítás módjára nem adnak választ. A közönséges differenciálegyenletek csak speciális  $f$  függvény esetén oldhatóak meg az ismert képletek segítségével. Ezért numerikus módszereket alkalmazunk a közelítő megoldás előállítására. Ez azt jelenti, hogy az értelmezési tartománynak egyes pontjaiban az ismeretlen megoldásfüggvény értékeit véges számú lépéssel közelítőleg határozzuk meg. A továbbiakban ilyen módszereket vizsgálunk. Ebben a fejezetben olyan eljárásokkal foglalkozunk, amelyek egy rögzített adott időpontbeli közelítést egy előző időpontbeli közelítés felhasználásával határoznak meg. Ezeket a módszereket egylépéses módszernek nevezzük.

Tehát a célunk a (2.2) Cauchy feladat egylépéses módszerekkel való közelítő megoldása, ahol  $T > 0$  mellett a fenti feladatnak létezik egyértelmű, megfelelően sima megoldása a  $[0, T]$  intervallumon.

Legyen  $\omega_h$  egy ekvidisztáns felosztású rácsháló, azaz

$$\omega_h = \{t_n = n \cdot h, n = 0, 1, \dots\} \text{ továbbá } h > 0.$$

Ekkor célunk olyan  $y_h : \omega_h \rightarrow \mathbb{R}$  rácshálós függvény megadása, amely jól közelíti  $u(t)$  pontos megoldást.

- jelölés: Legyen  $y_h(t_n) = y_n$   
Felmerülhet a kérdés, hogy mit jelent az, hogy  $y_n$  jól közelíti  $u(t_n)$ -t? Hogyan határozzuk meg  $y_h$  rácshálós függvényt? Elvárhatjuk, hogy ha olyan  $\omega_h$  rácshálót választunk, ahol  $h \rightarrow 0$  akkor minden  $t^* > 0$  rögzített pontra, ahol  $t^* \in \omega_h$  teljesül, hogy

$$\lim_{h \rightarrow 0} |y_n - \omega(t^*)| = 0, \quad n \cdot h = t^*$$

Ezt nevezzük konvergenciának.

## 2.3. Explicit Euler módszer

Tekintsük a következő numerikus módszert:

$$\frac{y_{n+1} - y_n}{h} = f(t_n, y_n), n = 0, 1, \dots \quad (2.3)$$

– Megjegyzés: Átrendezve az egyenletet a következőt kapjuk:

$$y_{n+1} = y_n + h \cdot f(t_n, y_n), n \in \mathbb{N}_0$$

Ekkor az algoritmus minden  $\omega_h$  rácshálózhoz meghatározza az  $y_h$  rácsfüggvényt.

Mivel  $y_n$  az ismeretlen  $u(t)$  függvény egy közelítése, ezért a kezdőpontban az értéke legyen  $y_0 = u(0) = u_0$  adott érték.

Ekkor a (2.3) összefüggést Explicit Euler-módszernek nevezzük.

**2.3.1. Definíció.** Az Explicit Euler módszer konvergencia, ha  $\lim_{h \rightarrow 0} |y_n - u(t^*)| = 0$  feltétel teljesül minden  $t^* \in [0, T]$ -re, ahol  $t^* = n \cdot h$ . Továbbá azt mondjuk, hogy a módszer  $p$ -ed rendű, ha

$$|y_n - u(t^*)| = O(h^p)$$

Azaz minél nagyobb  $p$  érték,  $y_n$  annál gyorsabban konvergál a megoldáshoz a rácsháló finomításával.

## 2.4. Az Explicit Euler-módszer hibafüggvénye

Legyen  $e_n = y_n - u(t_n)$  az  $\omega_h$  rácshálón értelmezett hibafüggvény. Jelölés: Legyen  $u_n = u(t_n)$

Ekkor átrendezve a fenti összefüggést, a következőt kapjuk:

$y_n = e_n + u_n$  Ezt felhasználva helyettesítsünk be  $y_n$  és  $y_{n+1}$ -et az  $\frac{y_{n+1} - y_n}{h} = f(t_n, y_n)$  képletbe:

$$\frac{e_{n+1} + u_{n+1} - (e_n + u_n)}{h} = f(t_n, e_n + u_n)$$

Rendezzük a fenti összefüggést:

$$\frac{e_{n+1} - e_n}{h} = f(t_n, e_n + u_n) - \frac{u_{n+1} - u_n}{h}$$

Ezt nevezzük a módszer hibaegyenletének.

Alakítsuk tovább a jobb oldalt, úgy, hogy bővítsük  $f(t_n, u_n)$ -el:

$$f(t_n, e_n + u_n) - f(t_n, u_n) + f(t_n, u_n) - \frac{u_{n+1} - u_n}{h}$$

Jelöljük  $f(t_n, e_n + u_n) - f(t_n, u_n)$  et  $\Psi_n^{(2)}$ -el, míg  $f(t_n, u_n) - \frac{u_{n+1} - u_n}{h}$ -et  $\Psi_n^{(1)}$ -vel. Tehát

$$f(t_n, e_n + u_n) - f(t_n, u_n) = \Psi_n^{(2)} = \Psi_n^{(2)}$$

$$f(t_n, u_n) - \frac{u_{n+1} - u_n}{h} = \Psi_n^{(1)}$$

Ekkor tehát a hibaegyenletet átírhatjuk a következő alakra

$$\frac{e_{n+1} - e_n}{h} = \Psi_n^{(1)} + \Psi_n^{(2)} \quad (2.4)$$

$\Psi_n^{(1)}$  azt mutatja meg, hogy a pontos megoldás milyen pontosan elégíti ki a numerikus megoldást meghatározó egyenletet.

Azaz  $\Psi_n^{(1)} = -\frac{u_{n+1}-u_n}{h} + f(t_n, u_n)$ , míg a numerikus megoldásra:  $0 = -\frac{y_{n+1}-y_n}{h} + f(t_n, y_n)$

Ezért  $\Psi_n^{(1)}$ -t reziduális hibának nevezzük, amely a numerikus módszer lokális approximációs hibája.

- Megjegyzés: Állítás, ha  $e_n \neq 0$ , akkor  $y_n \neq u_n$ . A fenti összefüggésből ez az állítás következik.

**2.4.1. Definíció.** Azt mondjuk, hogy a módszer approximálja a Cauchy feladatot, ha  $\lim_{h \rightarrow 0} \Psi_n^{(1)} = 0$ , továbbá  $p$ -ed rendben approximálja, ha  $\Psi_n^{(1)} = O(h^p)$

- Határozzuk meg az Explicit Euler-módszer lokális approximációs rendjét, ha  $u \in C^1(I)$ . Azaz határozzuk meg  $\Psi_n^{(1)}$ -t

$$\Psi_n^{(1)} = f(t_n, u_n) - \frac{u_{n+1} - u_n}{h} = f(t_n, u(t_n)) - \frac{u(t_n + h) - u(t_n)}{h}$$

Tudjuk, hogy definíció szerint  $f(t_n, u(t_n)) = u'(t_n)$ .

Fejtsük Taylor sorba  $u(t_n + h)$ -t  $t_n$  körül.

$$u(t_n + h) = u(t_n) + h \cdot u'(t_n) + \frac{h^2}{2} \cdot u''(t_n + h \cdot \nu)$$

ahol  $\nu \in (0, 1)$  szám.

Ezt behelyettesítve  $\Psi_n^{(1)}$  képletébe azt kapjuk, hogy

$$\begin{aligned} \Psi_n^{(1)} &= \frac{u(t_n) + h \cdot u'(t_n) + \frac{h^2}{2} \cdot u''(t_n + \nu h) - u(t_n)}{h} + u'(t_n) = \\ &= -\frac{h}{2} u''(t_n + \nu h) = O(h) \end{aligned}$$

Tehát az Explicit Euler-módszer első rendben approximálja a feladatot.

## 2.5. Az Explicit Euler módszer konvergenciája

Az Explicit Euler módszer:

$$\frac{y_{n+1} - y_n}{h} = f(t_n, y_n), \quad y_0 = u_0$$

és  $u'(t_n) = f(t_n, y_n)$  minden  $n$ -re. ahol  $y$  egy  $\omega_h \rightarrow \mathbb{R}$  rácsfüggvény, ahol  $\omega_h$  egy halmaz. Megjegyzés ha a  $[0, T]$  intervallumon értelmezett az  $y$  függvény és  $N$  az alappontok száma, akkor  $h = \frac{T}{N}$

**2.5.1. Definíció.** A módszer akkor konvergens, ha a hibafüggvénye tart 0-hoz. Hibafüggvény:  $e_n(h) = y_n - u(t^*)$ , ahol  $n \cdot h = t^*$  Tehát a konvergencia feltétele, hogy  $\lim_{h \rightarrow 0} e_n(h) = 0$

Megjegyzés: A módszer  $p$ -ed rendben konvergens, ha  $|e_n(h)| = O(h^p)$ .

Korábban beláttuk, a (2.4) összefüggést, azaz

$$\frac{e_{n+1} - e_n}{h} = \Psi_n^{(1)} + \Psi_n^{(2)},$$

ahol  $\Psi_n^{(1)}$  a lokális approximációs hiba,

míg  $\Psi_n^{(2)} = f(t_n, u(t_n) + e_n) - f(t_n, u(t_n))$

Az előző részben beláttuk, hogy  $|\Psi_n^{(1)}| \leq \frac{M_2}{2}h$  ahol  $M_2 = \max_{[0,T]} |u''|$ , azaz  $M_2$  a második derivált maximumát jelöli az intervallumon.

Feltehető, hogy  $f$  függvény második változójában lipschitzes (ahogy eddig).

$$|\Psi_n^{(2)}| = |f(t_n, u_n + e_n) - f(t_n, u_n)| \leq L|e_n|$$

ahol  $u_n = u(t_n)$  és  $L$  a Lipschitz állandó.

Tudjuk, hogy

$$|e_{n+1}| = |e_n + \Psi_n^{(1)} + \Psi_n^{(2)}| \leq |e_n| + |\Psi_n^{(1)}| + |\Psi_n^{(2)}| \leq$$

Behelyettesítve a fenti összefüggést:

$$\leq |e_n| + h|\Psi_n^{(1)}| + h \cdot L|e_n| = (1 + hL)|e_n| + h|\Psi_n^{(1)}|$$

Tehát, ha ismert az  $n$ -edik hibatag, akkor abból az  $n + 1$ -edik hibatagot meg tudjuk határozni.

Ezt a módszert folytatva tovább:

$$|e_n| \leq (1 + hL)|e_{n-1}| + h|\Psi_{n-1}^{(1)}|$$

$$|e_{n-1}| \leq (1 + hL)|e_{n-2}| + h|\Psi_{n-2}^{(1)}|$$

A fenti két összefüggésből felírhatjuk a következőt:

$$\begin{aligned} |e_n| &\leq (1 + hL)^2(|e_{n-2}| + h|\Psi_{n-2}^{(1)}|) + |\Psi_{n-1}^{(1)}| = \\ &= (1 + hL)^2|e_{n-2}| + ((1 + hL)\Psi_{n-2}^{(1)} + \Psi_{n-1}^{(1)})h \end{aligned}$$

Ezt az eljárást tovább folytatva az  $e_n$ -re tudunk  $e_0$  ismeretében becslést adni.

$$|e_n| \leq (1 + hL)^n |e_0| + h \sum_{j=0}^{n-1} (1 + hL)^j |\Psi_{n-1-j}^{(1)}|$$

Alkalmazzuk a következő becslést:

$$\sum_{j=0}^{n-1} (1 + hL)^j \leq (1 + hL)^n.$$

Ezt az állítást könnyen beláthatjuk, mivel

$$(1 + hL)^i < (1 + hL)^n, \quad i = 1 \dots n - 1$$

-re, mivel egy egynél nagyobb számot hatványozunk.

$$|e_n| \leq (1 + hL)^n (|e_0| + h \sum_{j=0}^{n-1} |\Psi_{n-1-j}^{(1)}|)$$

Vegyük észre, ha  $h \rightarrow 0$  akkor  $(1 + hL)^n \rightarrow 1$

**2.5.1. Tétel.**  $(1 + hL)^n \leq (e^{hL})^n$

*Bizonyítás.* Vizsgáljuk  $e^x$  hatványsorát:

$$e^x = 1 + x + \frac{x^2}{2} + \dots$$

$x > 0$  esetén, tehát  $e^x > 1 + x$  igaz lesz. Jelen esetben  $x = hL$  választással pont a kívánt egyenlőtlenséget kapjuk. Mivel  $h > 0$  és  $L > 0$ , azért  $x > 0$  feltétel is teljesül.

Tehát  $(1 + hL)^n \leq (e^{hL})^n = e^{n \cdot hL} = e^{t^* L}$  felhasználva, hogy  $t^* = n \cdot h$ .

Alkalmazva a becslést:

$$|e_n| \leq e^{t^*L} (|e_0| + h \cdot |\Psi_{n-1-j}^{(1)}|) \leq e^{t^*L} (|e_0| + h_{max} |\Psi_i^{(1)}| \cdot n)$$

Beláttuk, hogy  $|\Psi_i^{(1)}| \leq \frac{M_2}{2} h$  ezért:

$$|e_n| \leq e^{t^*L} (|e_0| + \frac{M_2 \cdot t^*}{2} h)$$

Tudjuk, hogy  $e_0 = y_0 - u_0 = 0$ , mert a kezdőpontban megköveteljük, hogy  $y_0 = u_0$  teljesüljön. Tehát bevezetve  $c = e^{t^*L} \frac{M_2 \cdot t^*}{2}$  jelölést, azt kapjuk, hogy  $|e_n| \leq c \cdot h$ .

Tehát  $\lim_{h \rightarrow 0} |e_n| = O(h)$

Ezzel beláttuk az állítást, tehát az Explicit Euler módszer konvergens.  $\square$

Megjegyzés: Vegyük észre, hogy ha  $c$  konstans értéke nagyon nagy, akkor  $h$  értékének nagyon kicsinek kell lennie, hogy a hiba a nullához tartson. Ez a gyakorlatban a számolást megnehezíti, mert ha  $h$  túl kicsi, ezért több számítógépes módszer nem találja meg.

- Megjegyzés: Az Explicit Euler módszer nem ekvidisztáns rácshálón is konvergens.

## 2.6. Implicit Euler-módszer

Az Implicit Euler-módszer egy ismert  $y_n$  értékből az  $y_{n+1}$  értéket minden egyes időlépésben egy egyenlet megoldásaként kapja meg.

Vizsgáljuk a következő numerikus módszert

$$\frac{y_{n+1} - y_n}{h} = f(t_{n+1}, y_{n+1}), \quad n = 0 \dots$$

- Megjegyzés: A fenti egyenletet átrendezve a következőt kapjuk:

$$y_{n+1} = y_n + h \cdot f(t_{n+1}, y_{n+1})$$

Az Implicit Euler-módszer esetén tehát minden lépésben egy algebrai egyenletrendszer oldunk meg.

**2.6.1. Tétel.** *A módszer lokális approximációs rendje:*

$$\begin{aligned} \Psi_n^{(1)} &= -\frac{u_{n+1} - u_n}{h} + f(t_{n+1}, u_{n+1}) = \\ &= -\frac{u(t_{n+1}) - u(t_{n+1} - h)}{h} + f(t_{n+1}, u_{n+1}). \end{aligned}$$

Fejtsük sorba a fenti kifejezésben szereplő  $u(t_{n+1} - h)$ -t  $t_{n+1}$  körül.

$$u(t_{n+1} - h) = u(t_{n+1}) - h \cdot u'(t_{n+1}) + \frac{h^2}{2} \cdot u''(t_{n+1} - h \cdot \nu)$$

ahol  $\nu \in (0, 1)$  szám.

Helyettesítsük be  $\Psi_n^{(1)}$  képletébe a fent kapott eredményt, továbbá használjuk fel, hogy definíció szerint  $u'(t_{n+1}) = f(t_{n+1}, u_{n+1})$

$$\begin{aligned}\Psi_n^{(1)} &= -\frac{u(t_{n+1}) - u(t_n) - h \cdot u'(t_n) + \frac{h^2}{2} \cdot u''(t_n)}{h} + u'(t_n) = \\ &= -\frac{h}{2} u''(t_n) = O(h^2)\end{aligned}$$

Tehát az Implicit Euler-módszer is elsőrendű.

- A további fejezetekben magasabb rendű módszereket keressünk.

## 2.7. A trapéz módszer

Tekintsük a következő numerikus módszert:

$$\frac{y_{n+1} - y_n}{h} = \frac{1}{2}(f(t_n, y_n) + f(t_{n+1}, y_{n+1}))$$

Ekkor, hasonlóan az előzőekhez, felírhatjuk a lokális approximáció rendjét:

$$\begin{aligned}\Psi_n^{(1)} &= -\frac{u(t_{n+1}) - u(t_n)}{h} + \frac{1}{2}(f(t_n, u(t_n)) + f(t_{n+1}, u(t_{n+1}))) = \\ &= -\frac{u(t_{n+1}) - u(t_n)}{h} + \frac{1}{2}(u'(t_n) + u'(t_{n+1}))\end{aligned}$$

Itt felhasználtuk, hogy

$$u'(t_n) = f(t_n, u(t_n))$$

és

$$u'(t_{n+1}) = f(t_{n+1}, u(t_{n+1}))$$

Fejtsük sorba  $u(t_{n+1}) = u(t_n + h)$ -t  $u(t_n)$  körül, illetve  $u'(t_{n+1}) = u'(t_n + h)$ -t  $u'(t_n)$  körül.

$$u(t_n + h) = u(t_n) + h \cdot u'(t_n) + \frac{h^2}{2} \cdot u''(t_n) + O(h^3) \text{ és}$$

$$u'(t_n + h) = u'(t_n) + h \cdot u''(t_n) + O(h^2)$$

Helyettesítsük be a fenti eredményeket  $\Psi_n^{(1)}$  képletébe:

$$\begin{aligned}\Psi_n^{(1)} &= \frac{u(t_n) + h \cdot u'(t_n) + \frac{h^2}{2} \cdot u''(t_n) + O(h^3) - u(t_n)}{h} + \\ &+ \frac{1}{2}(u'(t_n) + u'(t_n + h)) = u'(t_n) + h \cdot u''(t_n) + O(h^2)\end{aligned}$$

Egyszerűsítve a fenti kifejezést:

$$\Psi_n^{(1)} = O(h^2), \text{ tehát a trapéz módszer másodrendű.}$$

## 2.8. A $\Theta$ -módszer

Vizsgáljuk a következő paraméteres numerikus módszert:

$$\frac{y_{n+1} - y_n}{h} = \Theta f(t_{n+1}, y_{n+1}) + (1 - \Theta)f(t_n, y_n)$$



Ahol  $\Theta \in [0, 1]$  ekkor vegyük észre, hogy  $\Theta = 0$  esetén az Explicit Euler módszert,  $\Theta = 1$  esetén az Implicit Euler módszert, míg  $\Theta = \frac{1}{2}$  esetén a Trapéz módszert kapjuk.

Megjegyzés:  $\Theta \neq 0$  esetén a módszer implicit lesz, azaz minden lépésben egy egyenletet old meg  $y_{n+1}$  meghatározásához.

Megjegyzés: A módszer rendje  $\Theta = \frac{1}{2}$  esetén 2, míg minden egyéb esetben 1.

## 2.9. Magasabb rendű módszerek

Mi történik, ha nem csak a  $\omega_h$  rácsháló pontjaiban, hanem köztes pontokban is számolunk? Tehát két rácspont között vegyünk fel még tetszőleges számú segédpontot.

Például legyen  $t_{n+\frac{1}{2}}$  a  $t_n$  és  $t_{n+1}$  rácspontok között fél úton elhelyezkedő pont  $n \in \mathbb{N}_0$ -ra. (Ez tehát nem rácspont).

– Legyen tehát a felezőpontban definiálható numerikus módszer a következő:

$$\frac{y_{n+\frac{1}{2}} - y_n}{\frac{1}{2}h} = f(t_n, y_n) \quad (2.5)$$

Megjegyzés: Explicit Euler módszernél mindig az induló pontot kell nézni, míg Implicit Eulernél az érkező pontot.

$$\frac{y_{n+1} - y_n}{h} = f(t_{n+\frac{1}{2}h}, y_{n+\frac{1}{2}h}) \quad (2.6)$$

Számítsuk ki a fent definiált módszerek rendjét:

A (2.5) módszert átalakítva a következő összefüggést kapjuk:

$$y_{n+\frac{1}{2}} = y_n + \frac{1}{2}h \cdot f(t_n, y_n)$$

Ezt behelyettesítve a (2.6)-es képletbe:

$$\frac{y_{n+1} - y_n}{h} = f(t_{n+\frac{1}{2}h}, y_n + \frac{1}{2}h \cdot f(t_n, y_n))$$

összefüggést kapjuk.

Vizsgáljuk a fenti numerikus módszer rendjét:

**2.9.1. Tétel.** *A (2.5) módszer másodrendű.*

*Bizonyítás.* Látható, hogy

$$\Psi_n^{(1)} = -\frac{u(t_{n+1}) - u(t_n)}{h} + f(t_{n+\frac{1}{2}h}, u(t_n)) + \frac{1}{2}h \cdot f(t, u(t_n))$$

Tudjuk, hogy  $f(t_n, u(t_n)) = f(t_n)$ , továbbá

$\partial_1 f(t_n, u(t_n)) = \partial_1 f(t_n)$  és hasonlóan  $\partial_2 f(t_n, u(t_n)) = \partial_2 f(t_n)$ . Továbbá  $u''(t_n) = \partial_1 f(t_n, u(t_n)) + \partial_2 f(t_n, u(t_n)) \cdot f(t_n, u(t_n))$

Fejtsük sorba  $u(t_{n+1}) = u(t_n + h)$ -t  $t_n$  körül.

Ekkor a következőt kapjuk:  $u'(t_n) + \frac{h}{2}u''(t_n) + O(h^2)$ . Továbbá fejtsük sorba  $t_n$  körül  $f(t_{n+\frac{1}{2}h}, u(t_n)) + \frac{1}{2}h \cdot f(t, u(t_n))$ -et is:

$$f(t_n, u(t_n)) + \frac{1}{2}h \cdot \partial_1 f(t_n, u(t_n)) +$$

$$\begin{aligned}
& + \frac{1}{2}h \cdot f(t_n, u(t_n)) \cdot \partial_2 f(t_n, u(t_n)) + O(h^2) = \\
= & f(t_n, u(t_n)) + \frac{1}{2}h \cdot (\partial_1 f(t_n, u(t_n)) + f(t_n, u(t_n)) \cdot \partial_2 f(t_n, u(t_n))) + O(h^2) = \\
& = f(t_n, u(t_n)) + \frac{1}{2}h \cdot u''(t_n) + O(h^2)
\end{aligned}$$

ami felhasználva, hogy  $u'(t_n) = f(t_n, u(t_n))$  a következővel egyenlő:

$$u'(t_n) + \frac{1}{2}h \cdot u''(t_n) + O(h^2)$$

Tehát  $\Psi_n^{(1)}$  be behelyettesítve a sorbafejtéseket, azaz mind  $-\frac{u(t_{n+1})-u(t_n)}{h}$  helyére, mind  $f(t_{n+\frac{1}{2}h}, u(t_n)) + \frac{1}{2}h \cdot f(t_n, u(t_n))$  helyére behelyettesítve:

$$-(u'(t_n) + \frac{1}{2}h \cdot u''(t_n) + O(h^2)) + u'(t_n) + \frac{1}{2}h \cdot u''(t_n) + O(h^2) = O(h^2)$$

Tehát beláttuk, hogy a módszer másodrendű. □

– Megjegyzés:

$$\frac{y_{n+1} - y_n}{h} = f(t_{n+\frac{1}{2}h}, y_n + \frac{1}{2}h \cdot f(t_n, y_n))$$

módszer lépésszáma magas, mert mindent kétszer kell kiértékelni, azonban explicit módszer és másodrendű.

## 2.10. Runge–Kutta módszerek

A fenti ötletet alkalmazzuk, csak nem egy közbülső pontra, hanem többre:

Legyen:

$$\begin{aligned}
k_1 &= f(t_n, y_n) \\
k_2 &= f(t_n + a_2h, y_n + h \cdot b_{21}k_1) \\
k_3 &= f(t_n + a_3h, y_n + h \cdot (b_{31}k_1 + b_{32}k_2)) \\
&\vdots \\
k_s &= f(t_n + a_sh, y_n + h \cdot (b_{s1}k_1 + b_{s2}k_2 + \dots + b_{s,s-1}k_{s-1}))
\end{aligned}$$

Továbbá a numerikus módszer legyen a következő:

$$y_{n+1} = y_n + h \left( \sum_{i=1}^s \delta_i k_i \right)$$

Ahol  $a_2, a_3, \dots, a_s, b_{21}, \dots, b_{s,s-1}, \delta_1 \dots \delta_s$  adott paraméterek.

A fenti módszert  $s$  lépcsős Runge–Kutta módszernek nevezzük.

Az  $s$  lépcsős Runge–Kutta módszereket Butcher-táblával tudjuk ábrázolni:

$$\begin{array}{c|c} a & B \\ \hline & \delta \end{array}$$

Ahol  $a = 0, a_2, \dots, a_s$  oszlopvektor,  $B \in \mathbb{R}^{s \times s}$  mátrix és  $(B_{ij}) = b_{ij}$  továbbá  $\delta = \delta_1, \dots, \delta_s$  sorvektor.

A módszer explicit, ha  $B$  mátrix szigorúan alsó háromszögmátrix.

Az Explicit Euler-módszer Butcher táblója:

$$\begin{array}{c|c} 0 & 0 \\ \hline & 1 \end{array}$$

Mi a kapcsolat a rend és a lépcsőszám között?

Belátható, hogy a módszer rendje mindig kisebb, vagy egyenlő az  $s$  lépcsőszámmal.

## 3. fejezet

# Interpolációs módszerek alkalmazása a differenciálegyenletek numerikus megoldására

A következő fejezetben az előző részben megismert módszereket fogjuk alkalmazni egy ismert példára. A harmonikus rezgőmozgás egyenletét leíró differenciálegyenletet fogjuk megoldani numerikusan. Illetve vizsgálni fogjuk, hogy a numerikus megoldásból kapott pontokra interpolációt alkalmazva milyen pontos közelítést kapjuk az ismert megoldásfüggvénynek. A tesztfeladat tehát a harmonikus rezgőmozgást leíró ismert egyenlet numerikus módszerekkel való közelítése.

A fejezetben használt programkódokhoz a [5] oldalakat használtuk.

Az első fejezetben vizsgáltuk, ha az Explicit Euler módszerrel oldjuk meg a differenciálegyenletet akkor milyen pontos eredményt kapunk. Vizsgálni fogjuk, ha Implicit Euler módszerrel oldjuk meg az differenciálegyenletet, akkor milyen pontos lesz az eredmény, illetve azt is nézzük, hogy hogyan változik a megoldás, ha Csebisev alappontokon nézzük már magát a numerikus megoldását a differenciálegyenletnek. A differenciálegyenlet megoldásai pontpárok lesznek, így alkalmazhatunk rájuk interpolációt. Vizsgálni fogjuk, hogy a Lagrange módszerrel, a Newton módszerrel illetve a Spline interpolációval milyen pontosan kapjuk vissza az eredeti megoldás függvényét.

Célunk tehát a harmonikus rezgőmozgás függvényének minél pontosabb közelítése numerikus módszerekkel.

### 3.1. A harmonikus rezgőmozgás egyenlete

Ha rugóhoz rögzített testet kitérítünk nyugalmi helyzetéből és magára hagyjuk, akkor a test harmonikus rezgőmozgást végez. A testet pontszerűnek tekintjük, a külső erők hatásától eltekintjük.

A harmonikus rezgőmozgást leíró kezdetiérték feladat a következő:

$$\begin{cases} u''(t) = -\frac{D}{m}u(t), & t > 0, \\ u(0) = u_0, \\ u'(0) = v_0, \end{cases} \quad (3.1)$$

ahol  $u_0$  és  $v_0$  valós értékek adottak,  $D > 0$  a rugóállandó, azaz a rugóra jellemző direkción állandó,  $m > 0$  pedig a rugóra helyezett test tömege. Jelöljük  $\frac{D}{m}$ -et  $\omega^2$ -el. Vegyük észre, hogy  $\omega^2$  egy pozitív állandó. Jelen dolgozatban azt az esetet vizsgáljuk, amikor 1 az értéke.

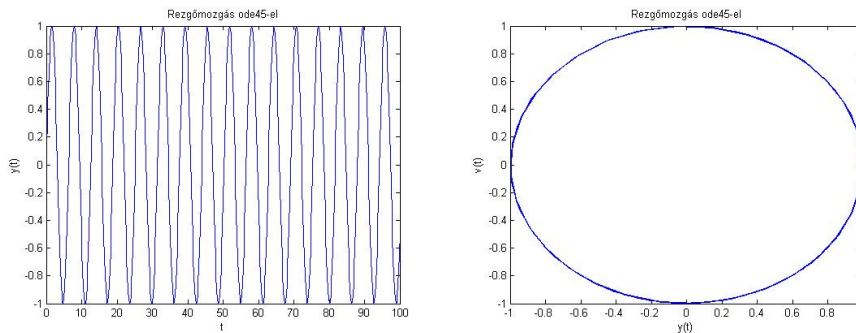
Vegyük észre, hogy ez egy másodrendű differenciálegyenlet, amit meg tudunk oldani elemi módszerek segítségével.

A harmonikus rezgőmozgást leíró másodrendű differenciálegyenlet megoldása:

$$\begin{aligned} u(t) &= u_0^2 \cdot (\sin^2(\omega(t)) + \cos^2(\omega(t))) + \\ &+ \frac{v_0^2}{\omega^2} (\sin^2(\omega(t)) + \cos^2(\omega(t))) = \\ &= u_0^2 + \frac{v_0^2}{\omega^2}. \end{aligned}$$

Alkalmazva a  $\sin^2(\omega(t)) + \cos^2(\omega(t)) = 1$  összefüggést.

Rajzoljuk ki a differenciálegyenlet pontos megoldását a MATLAB program segítségével. Alkalmazzuk a beépített *ode45* parancsot. Az *ode45* negyed és ötöd rendű Runge-Kutta módszert alkalmaz a megoldás előállításához. Egy adott pontban, egy kezdeti  $h$  értékkel kiszámítja a negyed és ötöd rendű módszerrel kapott eredmény különbségét. Amennyiben ez az érték nagyobb egy adott küszöb értéknél, új kezdeti  $h$  értéket választ, és ezzel újra számol, egészen addig amíg a negyed és ötöd rendű Runge-Kutta módszerrel kapott eredmények közel vannak egymáshoz. Az ilyen típusú módszereket beágyazott módszereknek nevezzük. Tehát az *ode45* egy változó lépésközű beágyazott Runge-Kutta módszer.



3.1. ábra. A (3.1) feladat megoldása a  $[0, 100]$  intervallumon. Az első ábra a rugó helyzetét mutatja az időben függvényében, míg a második ábrán a rugó sebességét láthatjuk a helyzetének függvényében.

Láthatjuk, hogy a 3.1 ábrán a rugó helyzete az időben egy  $\sin$  függvény. A rugó sebessége a helyzet függvényében pedig egy kör.

A differenciálegyenletet tehát a MATLAB megoldja az *ode45* segítségével, illetve alkalmaz egy belső interpolációs eljárást a kirajzoláshoz.

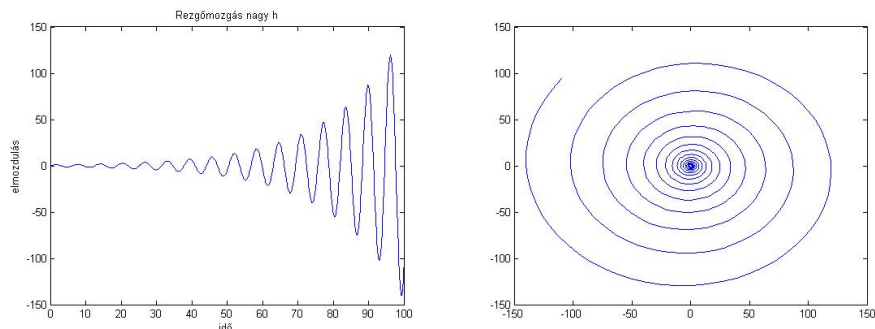
Vizsgáljuk, hogy ha az *ode45* helyett más numerikus módszert alkalmazunk a differenciálegyenlet megoldására, illetve ha a MATLAB beépített interpolációja helyett, a dolgozat első fejezetében mutatott interpolációs módszert alkalmazunk, akkor hogyan változik a megoldás pontossága.

## 3.2. Explicit Euler-módszer alkalmazása

Legyen  $h$  rögzített, azaz a felosztás ekvidisztáns.

Használjuk a programot a feladat megoldására. Legyen a megoldás érték készlete a  $[0, 100]$  intervallum.

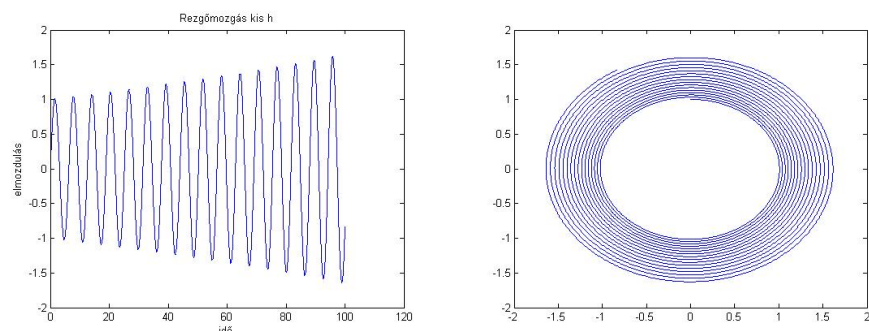
- Első lépésben válasszuk meg  $h$  értékét 0,1-nek. Ez numerikus módszer esetén viszonylag nagy  $h$ -nak minősül. Vizsgáljuk meg, hogy az Explicit Euler módszer nagy  $h$  esetén hogyan közelíti az eredeti  $\sin$  függvényt.



3.2. ábra. A (3.1) feladat numerikus megoldása az Explicit Euler-módszer segítségével,  $h = 0,1$  lépésköz választással. Az első ábrán a megoldást az időfüggvényében láthatjuk, míg a másodikon a fázistérben.

A 3.2 ábrán láthatjuk, hogy az Explicit Euler módszer elszáll a végtelenbe. A módszer nem stabil. Az alakja eleinte hasonlít a (3.1) pontos megoldáshoz, azonban a hullámszáma az ábrának növekszik és a végtelenhez tart.

- Második lépésben szintén az Explicit Euler módszert alkalmazzuk, de kisebb lépésközzel,  $h = 0,01$ -re. Vizsgáljuk, hogy ekkor az történik-e amit várunk, azaz a megoldás pontosabban közelíti az eredeti sin megoldást.



3.3. ábra. A (3.1) feladat numerikus megoldása az Explicit Euler-módszer segítségével,  $h = 0,01$  lépésköz választással. Az első ábrán a megoldást az időfüggvényében láthatjuk, míg a másodikon a fázistérben.

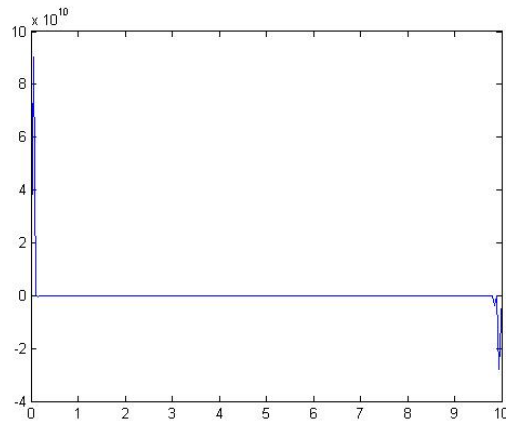
A 3.3 ábrán láthatjuk, hogy kis  $h$  esetén megoldás jobban közelíti a 3.1 pontos megoldást, mint a 3.2 ábra, azonban ez is elszáll a végtelenbe, igaz lassabban, mint azt nagy  $h$  esetén tette.

Láthatjuk, hogy mindkettő esetben a numerikus megoldás elszáll, nem tart az eredeti pontos megoldáshoz. Kis  $h$  esetén lassabban száll el a végtelenbe. Láthatjuk, hogy a módszer kis  $h$  esetén lesz pontosabb.

### 3.3. Lagrange interpoláció alkalmazása a numerikus megoldásra

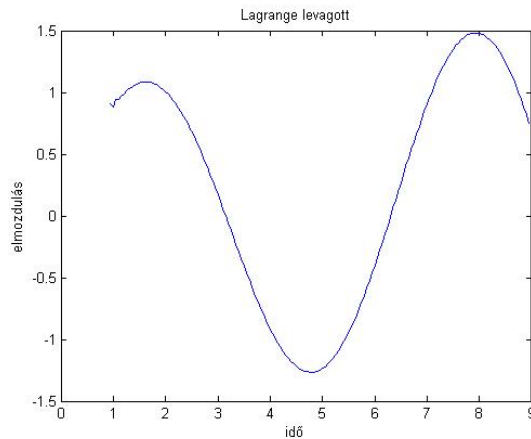
Az első fejezetben beláttuk, hogy a Lagrange interpoláció kis  $h$  esetén lesz pontos, ezért vizsgáljuk csak ebben az esetben a numerikus megoldást az Explicit Euler-módszerrel.

Láthatjuk az ábrán, hogy a megoldás a kezdő illetve végpontok környékén elszáll. A könnyebb áttekinthetőség kedvéért vizsgáljuk a numerikus megoldást a  $[0, 10]$  intervallumon.



3.4. ábra. A (3.1) feladat Explicit Euler-módszerrel számolt numerikus megoldására alkalmazott Lagrange polinom. Az  $u(t)$  megoldást a  $t$  függvényében ábrázoljuk  $h = 0,1$  esetén.

A 3.4 ábrán láthatjuk, hogy az intervallum kezdő, illetve végpontjában a megoldás elszáll. Vizsgáljuk, hogyan viselkedik az intervallum belsejében? Vegyük észre, hogy a megoldást a  $[t_{20}, t_{180}]$  alappontok által meghatározott intervallumon jól közelíti.



3.5. ábra. A 3.1 feladat Explicit Euler-módszerrel számolt numerikus megoldására alkalmazott Lagrange polinom a  $[t_{20}, t_{180}]$  intervallumon.

A 3.5 ábrán láthatjuk, hogy az intervallum belsejében a numerikus módszer jól közelíti a megoldást.

Vegyük észre, hogy a differenciálegyenlet numerikus megoldása akkor közelítette jól a megoldást, ha  $h$  értéke kicsi. Ebben az esetben ha a kapott megoldás pontokra interpolációt alkalmazunk, akkor kis  $h$  esetén sok alappont lesz. Sok alappont esetén az interpolációs polinom magas fokú lesz. A magasfokú polinom azonban a szomszédos alappontok között nem feltétlen megfelelően simán halad át. Ez pontatlanságot eredményezhet.

Tehát láthatjuk, hogy az Explicit Euler módszer instabilitása is okozhatja a megoldás pontatlanságát (elszáll a végtelenbe a megoldás). Tudjuk, hogy az Implicit Euler módszer jobb stabilitási tulajdonságokkal rendelkezik (A-stabil), ezért alkalmazzuk a differenciál egyenlet numerikus megoldására.

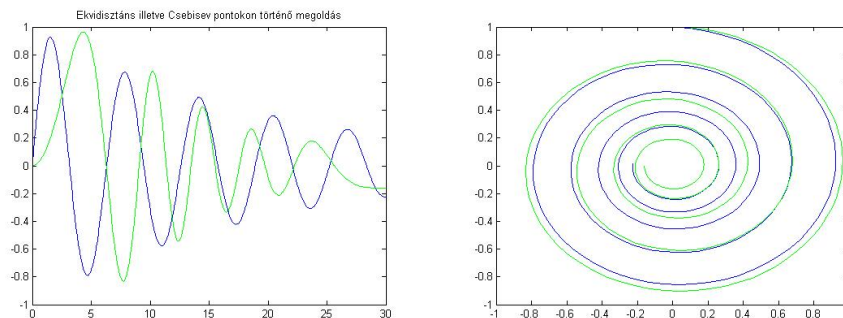
### 3.4. Lagrange interpoláció Csebisev alappontokon

Az Implicit Euler módszert vizsgáljuk ekvidisztáns alappontokra és Csebisev alappontokra is. Nézzük meg, hogy melyik esetben kapunk pontosabb megoldást. A 3.1 ábrán láthatjuk, hogy a pontos megoldás a fázistérben egy kör, ezért nézzük meg, azt is, hogy melyik közelíti jobban a pontos megoldást (kört).

Láthatjuk, hogy mindkettő módszer eleinte közel halad a pontos megoldáshoz, a Csebisev alappontokon történő megoldás sokáig közelebb marad a pontos megoldáshoz. Azonban mindkettő megoldás befelé spirálozva a 0-hoz tart.

Ekvidisztáns felosztás esetén ez a módszer sem eredményez pontos megoldást. Vegyük észre, hogy a megoldás megint a szélén száll el. Sűrítjük a felosztást a széleken, tehát vegyük Csebisev alappontokon az Implicit Euler módszert.

A könnyebb áttekinthetőség kedvéért a megoldást a  $[0, 30]$  intervallumon ábrázoljuk.



3.6. ábra. Az ábrán láthatjuk a Csebisev alappontokon (kék), illetve az ekvidisztáns alappontokon (zöld), az Implicit Euler-módszer megoldásaira interpolált Lagrange polinomot.

A 3.6. ábrán láthatjuk, hogy ekvidisztáns felosztás esetén gyorsabban tart nullához, azaz Csebisev alappontokon pontosabb megoldást kapunk. Azonban a megoldás nem száll el a végtelenbe, mint az Explicit módszer esetén.

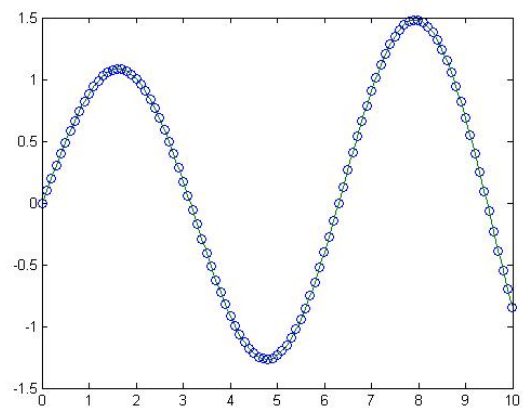
Ez a numerikus megoldás sem közelíti megfelelően a megoldást. Tehát az alappontok finomításával vagy sűrítésével sem tudunk kellően pontos numerikus megoldást előállítani. Vizsgáljuk, ha az alappontok ekvidisztánsak, de több információ adott a függvényről, azaz bizonyos rendben a deriváltak is adottak. A Spline interpolációnál az alappontokban a derivált értékek is adottak.

### 3.5. Spline interpoláció

A könnyebb átláthatóság kedvéért a megoldást ábrázoljuk a  $[0, 10]$  intervallumon.

A 3.7 ábrán láthatjuk, hogy a Spline interpoláció a kezdő illetve a végpontokban is jól közelíti a megoldást, illetve áthalad az adott pontokon.

A MATLAB beépített Spline interpolációs függvényét használjuk. A beépített függvény kvadratus Spline interpolációt alkalmaz, lásd [6]. Ez a megoldás tűnik a legpontosabbnak. Az ábrán láthatjuk, hogy teljesíti az (1.1) interpolációs alapfeltételt, azaz átmegy az adott pontokon.



3.7. ábra. A (3.1) feladatra alkalmazott Implicit Euler-módszer megoldásai (kék körök) és a rájuk alkalmazott Spline interpoláció eredménye (zöld görbe) az idő függvényében.



# Összefoglalás

A dolgozatban közönséges differenciálegyenlet kezdeti érték feladat megoldásából származó pontokra alkalmazott interpolációs módszereket vizsgáltunk. Láthattuk, hogy a differenciálegyenlet numerikus megoldására alkalmazott módszer nagyban befolyásolta az eredmény pontosságát. Láthattuk, hogy ha az Explicit Euler-módszert használjuk, a megoldás elszáll a végtelenbe. Továbbá, ha a lépésköz ( $h$ ) értékét kicsinek választottuk, a megoldás pontosabb lett.

Láthattuk, hogy az interpoláció akkor pontos, ha kis  $h$  értéket választunk, míg a differenciálegyenlet megoldására alkalmazott numerikus módszerek kevés pont esetén pontosak. A két feltétel együtt sajnos nem teljesül, így kaphattunk olyan eredményeket, ahol a megoldás a kezdeti és végpontokban elszáll. Mivel a megoldás csak a kezdő és végpontok környékén nem stimmelt, ezért alkalmaztuk a Csebisev alappontokon történő interpolációt. Az alappontok speciális megválasztásával a kezdő és végpontokban több adott pontunk lett, így azokon a tartományokon is pontosabb megoldást kaptunk.

Mivel a Csebisev alappontokon történő numerikus megoldás sem adott pontos megoldást, ezért alkalmaztuk a Spline interpolációt a megoldás ábrázolására. A Spline interpoláció kiküszöbölte a megoldás eddigi nagy hullámzását a kezdő és végpontok környékén, így nagy pontosságú megoldást kaptunk.

# Irodalomjegyzék

- [1] J. R. Chasnov: Introduction to Numerical Methods, The Hong Kong University of Science and Technology, egyetemi jegyzet (2012)
- [2] Faragó I., Horváth R.: Numerikus módszerek, TYPOTEX, egyetemi jegyzet (2013)
- [3] J. Stoer, R. Bulirsch: Introduction to Numerical Analysis, Springer-Verlag (1991)
- [4] T. Young and M. J. Mohlenkamp, Introduction to Numerical Methods and Matlab Programming for Engineers, Ohio University, egyetemi jegyzet (2017)
- [5] <https://www.mathworks.com/products/matlab.html>
- [6] <https://www.mathworks.com/help/matlab/ref/spline.html>

# Függelék

Pontos megoldás el állítása:

```
tspan=[0 100]
y0=[0; 1];
[t,y]=ode45(@diffeq(t,y), tspan, y0)
plot(t, y(:, 1))
title('Rezg mozgás ode45-el ');
xlabel('t');
ylabel('y(t)');
plot(y(:, 1), y(:, 2))
title('Rezg mozgás ode45-el ');
xlabel('y(t)');
ylabel('v(t)')
```

Az Explicit Euler-módszer kódja a MATLAB programban:

```
function [t,y]=explicituler_rendszer(diffeq, t0, y0, h, T)

t=zeros(N+1, 1);
y=zeros(2, N+1);
t(1)=t0;
y(:, 1)=y0;
for i=1:N
    t(i+1) = t(i) + h;
    y(:, i+1) = y(:, i) + h * diffeq(t(i), y(:, i));
end
```

Explicit Euler módszer futtatása kis  $h$  esetén.

```
clear all
t0=0;
y0=[0; 1];
T=100;
h=0.01;
[teek, yeek]=explicituler_rendszer(@diffeq, t0, y0, h, T);
plot(teek, yeek(1, :))
title('Rezg mozgás kis h ');
xlabel('id ');
ylabel('elmozdulás');
```

Explicit Euler módszer nagy  $h$  esetén.

```
clear all
t0=0;
y0=[0; 1];
T=100;
h=0.1;
[teen, yeen]=expeuler_rendszer(@diff, t0, y0, h, T);
plot(teen, yeen(1, :))
title('Rezg mozgás nagy h');
xlabel('id');
ylabel('elmozdulás');
```

Lagrange interpoláció kódja:

```
function aaa=Lagrange2(x, pointx, pointy)
n=size(pointx, 1);
L=ones(n, size(x, 2));
for i=1:n
    for j=1:n
        if (i~=j)
            L(i, :)=L(i, :).*(x-pointx(j))/(pointx(i)-pointx(j));
        end
    end
end
aaa=0;
for i=1:n
    aaa=aaa+pointy(i)*L(i, :);
end+

x=(0:0.05:10);
pointx=teen;
pointy=yeen(1, :);
aaa=Lagrange2(x, pointx, pointy) %vektor dim ek mi att.

plot(x, aaa)
axis([0 10 -2 2])
title('Lagrange interpol nagy h');
xlabel('id');
ylabel('elmozdulás');
```

Lesz kített intervallumon a Lagrange interpoláció.

```
plot(x(20:180), aaa(20:180))
```

Explicit Euler módszer Csebi sev alappontokon:

```
function [t, y]=expeuler_rendszer_csebi sev(diff, y0, N)

a=0;
```

```

b=10;
t = 0.5*(a+b)+0.5*(b-a)*cos( (2 * (1:N) - 1) / (2 * N) * pi);
x=sort(t);
h=zeros(1, N+1);
y=zeros(2, N);

y(:, 1)=y0;
for j=2: N
    h(j-1)=x(j)-x(j-1);
end
for i=1: N-1
    y(:, i+1) = y(:, i) + h(i) * diff(x(i), y(:, i));
end

```

```
clear all
```

```

N=10;
y0=[0; 1];
a=0;
b=10;
[tcs, ycs]=expeuler_rendszer_csebi_sev(@diff, y0, N);
plot(tcs, ycs(1, :))
%axis([0 10 -2 2])
title('Rezg mozgás nagy h');
xlabel('id');
ylabel('elmozdulás');

```

Implicit Euler módszer:

```

a = 0; b = 30;
n = 300;
t = 0.5*(a+b)+0.5*(b-a)*cos( (2 * (1:n) - 1) / (2 * n) * pi);
tt=sort(t);
y=zeros(2, n);
y(:, 1)=[0; 1]; %y0
for i=1: n-1
    h(i)=tt(i+1)-tt(i);
end
A=[0, 1; -1 0];

for j=2: n
    y(:, j)=(inv(eye(2)-h(j-1)*A))*y(:, j-1);
end
plot(tt, y(1, :))

```

```

a = 0; b = 30;
n = 300;

```

```

tt=linspace(a, b, n);
y=zeros(2, n);

```

```

y(:, 1)=[0; 1];
for i=1: n-1
    h(i)=tt(i+1)-tt(i);
end
A=[0, 1; -1 0];

for j=2: n
    y(:, j)=(i nv(eye(2)-h(j-1)*A))*y(:, j-1);
end
plot(tt, y(1, :))

yc1=y(1, :);
yc2=y(2, :);
ye2=y(2, :);
ye1=y(1, :);

plot(ye1, ye2, ' b', yc1, yc2, ' g', ypontos(:, 2), ypontos(:, 1), ' r')
title(' Implicit Euler megoldások')

tte=tt;
ttc=tt;
plot(tte, ye1, ' b', ttc, yc1, ' g', tpontos, ypontos(:, 1), ' r')

x=(0: 0. 05: 30);
pointx=ttc;
pointy=yc1(1, :);
aaa=Lagrange2(x, pointx, pointy)

```