EÖTVÖS LORAND
TUDOMÁNYEGYETEM

TERMÉSZETTUDOMÁNYI KAR

Közelítő integrálás interpolációs formulákkal

Készítette:
Molnár Roland
Matematika BSC
Elemző szakirány

Témavezető:
Dr. Csomós Petra
Alkalmazott Analízis és Számításmatematikai Tanszék

2022

NYILATKOZAT

Név: Molnár Roland

ELTE Természettudományi Kar, szak: Matematika BSC

NEPTUN azonosító: EBOU2I

Szakdolgozat címe:
Közelítő integrálás interpolációs formulákkal

A **szakdolgozat** szerzőjeként fegyelmi felelősségem tudatában kijelentem, hogy a dolgozatom önálló szellemi alkotásom, abban a hivatkozások és idézések standard szabályait következetesen alkalmaztam, mások által írt részeket a megfelelő idézés nélkül nem használtam fel.

Budapest, 2022.05.31.

Molnár Roland

a hallgató aláírása

Köszönetnyílvánítás

Szeretném megköszönni a családomnak és a barátaimnak a támogatást, amit nyújtottak a szakdolgozat készítése közben. Külön köszönetet szeretnék mondani a témavezetőmnek, aki kitartó volt és rengeteget segített, hogy a szakdolgozat jelenlegi formájában legyen.

Tartalomjegyzék

1. Bevezetés

1.1. Témaválasztás

Ezt a témát azért választottam, mert lehetőségeket láttam olyan módszerek felfedezésére, amik később még hasznosak lehetnek a számomra, illetve a programozás is szempont volt a témám választásánál.
A szakdolgozatban a közelítő integrálás különböző módszereivel fogok foglalkozni, aminek több különféle fajtája van, és mindegyik fajtájának van valami előnye és hátránya is az alkalmazásban.
Ezek a módszerek egy függvényhez tartozó ponthalmazhoz rendelnek hozzá egy olyan függvényt, amelyet minden esetben egyszerű integrálni, és ezzel egy közelítést kapni az eredeti függvény integráljának az értékéhez.

1.2. Közelítő integrálás

Az alapprobléma az, hogy adott egy ponthalmazunk függvényvértékekkel, és ki szeretnénk számolni az eredeti függvény görbe alatti területét egy adott $[a, b]$ intervallumon. A probléma az, hogy általában nem ismerjük magát az $f(x)$ függvényt, csak néhány pontjában vett értéket, vagy ha ismerjük is, túl bonyolult lenne integrálni, esetleg nem is lehet kiszámolni az integráljának az értékét. Az ilyen esetekben jön szóba a közelítő integrálás, ahol egy $p(x)$ interpolációs formula integrálásával szeretnénk megadni az $f(x)$ függvény integráljának a közelítését, azaz $f(x)$ függvényt közelítjük $p(x)$-szel.

1. Definíció. *Jelöljük $p(x)$-szel azt az interpolációs formulát, aminek az integrálja az $[a, b]$ intervallumon közelíti az $f(x)$ függvény $[a, b]$ intervallumon vett integráljának az értékét:*

$$\int_a^b p(x)\,dx \approx \int_a^b f(x)\,dx.$$

Az interpolációs formuláktól két fontos dolgot várunk: hogy elég pontos eredményt adjanak és hogy rövid számítási idővel rendelkezzenek. A továbbiakban ilyen interpolációs formulákkal fogunk foglalkozni és összehasonlítjuk, hogy mennyire használhatóak bizonyos feltételek mellett.

2. Lagrange-interpoláció

2.1. Bevezetés

Az első interpolációs formulánál, amit vizsgálunk, az az alapötlet, hogy az adott ponthalmazra interpolálunk egy polinomot, és ezt az egyszerűen integrálható függvényt integráljuk az adott intervallumon, amivel kapunk egy közelítést az eredeti függvény integráljára az adott intervallumon.

2.2. Egyszerű módszer

2. Definíció. *A Lagrange-interpoláció az az eljárás, ahol adott n darab $(x_i, y_i) \in [a, b]$ ponthoz rendelünk egy $n-1$ fokszámú polinomot, ami átmegy az adott pontokon. Ezt a polinomot jelöljük $L(x)$-szel.*

Az eljárás során az $\ell_i(x)$ Lagrange polinomok, azaz $L(x_i) = y_i$, $(i = 1, \ldots, n)$ összeadásával kapjuk meg az úgynevezett Lagrange-interpolációs formulát, amit integrálva megkapjuk a keresett $f(x)$ integrál értéket.
Az $\ell_i(x)$ polinomok kiszámítása:

$$\ell_i(x) = \frac{x - x_1}{x_i - x_1} \cdot \frac{x - x_2}{x_i - x_2} \cdot \ldots \cdot \frac{x - x_{i-1}}{x_i - x_{i-1}} \cdot \frac{x - x_{i+1}}{x_i - x_{i+1}} \cdot \ldots \cdot \frac{x - x_n}{x_i - x_n}.$$

Ezt a polinomot átírva egyszerűbb alakra kapjuk a következőt:

$$\ell_i(x) = \prod_{\substack{k=1 \\ k \neq i}}^{n} \frac{x - x_k}{x_i - x_k}. \tag{1}$$

Az $\ell_i(x)$ polinomokat összeadva és megszorozva az y_i függvényértékekkel kapjuk $L(x)$ alakját:

$$L(x) = \sum_{i=1}^{n} \ell_i(x) \cdot y_i = \sum_{i=1}^{n} \left(\prod_{\substack{k=1 \\ k \neq i}}^{n} \frac{x - x_k}{x_i - x_k} \right) \cdot y_i. \tag{2}$$

Látható, hogy $L(x_i) = y_i$.
Az $L(x)$ polinom integrálásával az $[a, b]$ intervallumon kapunk egy közelítést az eredeti függvény integráljára:

$$\int_a^b f(x)\, dx \approx \int_a^b L(x)\, dx.$$

2.3. Programkóddal kapott eredmények

A programkódok amiket megemlítek, megtalálhatóak a Függelékben. Az összes függvényt a [0,1] intervallumon nézzük. A programkódot az l11(s2(),s1()) parancsal futtattam, ahol az s2.m adja meg a függvényt, és az s1.m adja meg a [0,1] intervallumon elhelyzkedő pontokat. A Lagrange-interpoláció hibáját úgy kaptuk meg, hogy az l11.m-ben lefutott a Lagrange-interpoláció integráljának az eredménye és az s3.m segítségével határoztam meg a pontos integrál eredményét és ennek a kettőnek a különbségének az abszolútértéke lett a Lagrange interpoláció hibája.

Függvény	Pontszám	Lagrange hibája
$3x^4 + 3x^3 + 5x^2 + 2x$	169	1,1842e+135
$10x^6 + 3x^5 + x^4 + 8x^3 + 4x^2 + 6x + 2$	77	3,5391e+52
$7x^2 + 3x + 3$	170	1,9692e+138
$x^2 + 7x + 2$	280	2,0968e+241
4	280	3,0443e+236
$5x^7 + 8x^6 + 2x^5 + 7x^4 + 9x^3 + 2x^2 + 4x$	226	2,1416e+193
$10x^2 + 10x + 10$	158	6,1562e+121
$4x^2 + x + 1$	72	5,1750e+43
$x^3 + 2x + 3$	66	5,2115e+39
$8x^4 + 8x^3 + 6x^2 + 4x + 9$	251	6,9365e+217

A hibák hatalmasak lettek minden esetben és a függvény változtatása elenyésző mértékben befolyásolta a hiba növekedését. Jól látható, hogy minél több pontot veszünk, a Lagrange-interpolációval kapott közelítés annál nagyobb hibát ad, tehát a következő részben részintervallumokon használjuk a Lagrange-interpolációt, hogy sokszor kevés ponton végezzük el a formulát, és ezzel talán nagyobb pontosságot érünk el.

2.4. Összetett módszer

Az előbb láttuk, hogy elég pontatlan a módszer, amikor az egész ponthalmazra akarjuk használni a Lagrange-interpolációt. Nézzük meg, milyen eredményeket ad, ha különböző nagyságú részintervallumokra használjuk azt, és figyeljük meg, hogy melyik módszert éri meg alkalmazni a későbbi számításokhoz, ha megállapítható olyan részintervallumszám, amire a Lagrange-interpolációt alkalmazva nagy százalékban elfogadható eredményt ad.

3. Definíció. *Az összetett módszer úgy áll elő, hogy N részintervallumon alkalmazzuk az egyszerű módszert és a kapott eredményeket összeadjuk. Az egyszerű módszer eredményét egy adott intervallumon jelöljük $L_i(x)$-el,ekkor*

$$L(x) = \sum_{i=1}^{N} L_i(x).$$

2.5. Programkóddal kapott eredmények

A programkódok amiket megemlítek, megtalálhatóak a Függelékben. A következő formulákat a [0,1] intervallumon vizsgáljuk. A kod5.m programot használjuk a formulák és az eredeti integrál különbségének meghatározására. Az integrál pontos értékét az s3.m kóddal kaptuk meg, a függvényt az s2.m programmal generáltuk, míg a pontokat az s1.m programmal generáltuk. Az alsó táblázat azt mutatja, hogy a Lagrange-interpolációt hány ponton végeztük el egy részintervallumban.

Függvény	Pontszám
$6x^5 + x^4 + 9x^3 + 7x^2 + 7x + 1$	860
$6x^5 + 2x^4 + 8x^3 + 2x^2 + 4x + 9$	856
$6x^7 + 9x^5 + 5x^4 + 5x^3 + 6x^2 + 8x + 9$	383
$7x^2 + 7x + 5$	715
$3x^4 + 3x^3 + 5x^2 + 2x$	169
$6x^3 + 2x^2 + 6x + 7$	601
$9x + 8$	710
$10x^6 + 3x^5 + x^4 + 8x^3 + 4x^2 + 6x + 2$	77
$7x^2 + 3x + 3$	170
$x^2 + 7x + 2$	280

2 pontra	3 pontra	4 pontra	5 pontra
4,0205e+03	3,9903e+07	2,6368e+03	4,4838e+12
8,5569e+03	3,7835e+07	3,9479e+03	1,4107e+13
4,0124e+03	4,5271e+03	1,8304e+03	3,6845e+06
6,9006e+03	3,8567e+03	2,5675e+03	1,0024e+10
234,7953	359,0018	209,6621	164,6828
6,1619e+03	3,6421e+03	2,4212e+03	1,8164e+03
8,8500e+03	2,2867e+06	2,9374e+03	2,2055e+03
200,0751	529,3438	160,5597	203,8330
952,7344	3,1413e+04	375,8333	280,1671
1,5762e+03	7,6769e+04	536,6667	6,8507e+07

A táblázat alapján a 4 pontra használt Lagrange-interpoláció mindig jobb eredményt ad, mint a 2 és 3 pontra alkalmazott, tehát velük nem fogunk foglalkozni többet. Az 5 pontra alkalmazott Lagrange-interpoláció képes a legjobb eredményt adni a felsoroltak közül, de képes nagyon rossz eredményt is visszaadni. Összességében a vizsgáltak közül a 4 pontra alkalmazott Lagrange-interpoláció a legmegbízhatóbb, mert nem ad kiugróan rossz eredményt és mindig közel van a legjobb közelítéséshez.

3. Newton–Cotes formulák

3.1. Bevezetés

Jelen fejezetben az előzőnél szerteágazóbb témával fogok foglalkozni. A Newton–Cotes formulák két nagy csoportba oszthatók: zárt és nyitott formulák. A továbbiakban nézett formulák merőben különbözni fognak a Lagrange-interpolációnál látottaknál, mivel a Lagrange-interpolációnál egy polinomot kerestünk, aminek kiszámoltuk az integrálját, és ezzel kaptunk közelítést az eredeti integrálra, míg a mostani részben a használt formula eredménye fogja visszaadni az eredeti integrál közelítését, és nem kell integrálunk közben.

3.2. Newton–Cotes formulák fajtái

4. Definíció. *Zárt Newton–Cotes formuláról van szó, ha az adott n darab $(x_i, y_i) \in [a, b]$ esetén $a = x_0$ és $b = x_n$, tehát a formula magában foglalja a ponthalmaz végpontjait.*

5. Definíció. *Nyitott Newton–Cotes formuláról van szó, ha az adott n darab* $(x_i, y_i) \in [a, b]$ *esetén* $a \neq x_0$ *vagy* $b \neq x_n$*, tehát a formula nem foglalja magában a ponthalmaz végpontjait.*

3.3. Zárt Newton–Cotes formulák

A zárt Newton–Cotes formulákhoz legalább két pont szükséges, mivel magában kell foglalnia az intervallum végpontjait. A legelső ilyen formula a trapézformula, ennek levezetjük a képletét.

1. Tétel. *A trapézformula képlete* $\frac{1}{2} \cdot h \cdot (y_1 + y_2)$*, ahol* $h = x_2 - x_1$*.*

Bizonyítás. Használjuk a Lagrange-interpolációt két pontra:

$$L_2(x) = \frac{x - x_2}{x_1 - x_2} \cdot y_1 + \frac{x - x_1}{x_2 - x_1} \cdot y_2 = \frac{x - x_1 - h}{-h} \cdot y_1 + \frac{x - x_1}{h} \cdot y_2 =$$

$$= \frac{x}{h} \cdot (y_2 - y_1) + \left(y_1 + \frac{x_1}{h} \cdot y_1 - \frac{x_1}{h} \cdot y_2 \right).$$

Az $L_2(x)$-et integrálva a megfelelő intervallumon visszakapjuk a két pont közötti függvény integráljának az értékét:

$$\int_{x_1}^{x_2} f(x)\,dx = \int_{x_1}^{x_1+h} L_2(x)\,dx =$$

$$= \frac{1}{2 \cdot h} (y_2 - y_1) \cdot \left[x^2\right]_{x_1}^{x_2} + \left(y_1 + \frac{x_1}{h} \cdot y_1 - \frac{x_1}{h} \cdot y_2 \right) \cdot [x]_{x_1}^{x_2} =$$

$$= \frac{1}{2 \cdot h} (y_2 - y_1) \cdot (x_2 + x_1) \cdot (x_2 - x_1) + \left(y_1 + \frac{x_1}{h} \cdot y_1 - \frac{x_1}{h} \cdot y_2 \right) \cdot (x_2 - x_1) =$$

$$= \frac{1}{2} \cdot (y_2 - y_1) \cdot (2 \cdot x_1 + h) + y_1 \cdot h + x_1 \cdot (y_1 - y_2) =$$

$$= x_1 \cdot (y_2 - y_1) + \frac{1}{2} \cdot h \cdot (y_2 - y_1) + h \cdot y_1 - x_1 \cdot (y_2 - y_1) =$$

$$= \frac{1}{2} \cdot h \cdot (y_1 + y_2).$$

□

2. Tétel. *A Simpson formula képlete* $\frac{1}{3} \cdot h \cdot (y_1 + 4 \cdot y_2 + y_3)$*, ahol a* $h = x_2 - x_1$ *és* $x_2 - x_1 = x_3 - x_2$*.*

Bizonyítás. Valamely $A, B, C \in \mathbb{R}$ mellett vezessük be a következő jelölést: $f(x_i) = A \cdot x_i^2 + B \cdot x_i + C$ ahol, $i = 1, 2, 3$. Ekkor

$$\int_{x_1}^{x_3} f(x)\,dx = \int_{x_1}^{x_3} (A \cdot x^2 + B \cdot x + C)\,dx =$$

$$= \left(\frac{A}{3} \cdot x^3 + \frac{B}{2} \cdot x^2 + C \cdot x\right)_{x_1}^{x_3} = \frac{A}{3} \cdot (x_3^3 - x_1^3) + \frac{B}{2} \cdot (x_3^2 - x_1^2) + C \cdot (x_3 - x_1) =$$

$$= \frac{A}{3} \cdot (x_3 - x_1) \cdot (x_3^2 + x_3 \cdot x_1 + x_1^2) + \frac{B}{2} \cdot (x_3 - x_1) \cdot (x_3 + x_1) + C \cdot (x_3 - x_1) =$$

$$= \frac{x_3 - x_1}{6} \cdot \big(2 \cdot A \cdot (x_3^2 + x_3 \cdot x_1 + x_1^2) + 3 \cdot B \cdot (x_3 + x_1) + 6 \cdot C\big) =$$

$$= \frac{h}{3} \cdot (f(x_3) + f(x_1) + A \cdot (x_3 + x_1)^2 + 2 \cdot B \cdot (x_3 + x_1) + 4 \cdot C) =$$

$$= \frac{h}{3} \cdot (f(x_3) + f(x_1) + A \cdot (2 \cdot x_2)^2 + 2 \cdot B \cdot (2 \cdot x_2) + 4 \cdot C) =$$

$$= \frac{h}{3}(f(x_3) + 4 \cdot f(x_2) + f(x_1)) = \frac{h}{3} \cdot (y_1 + 4 \cdot y_2 + y_3)$$

tehát kapjuk az állításban szereplő formulát.

□

Felsorolok még pár zárt Newton–Cotes formulát, amiket nem vezetek le, mert nem fogok velük foglalkozni a későbbiekben.
Az alábbi táblázatban a pontszámok a formulák által használt pontok számát jelölik, tehát mondjuk a Simpson formulát egy ponthalmazon használva úgy tehetjük meg, ha kiválasztunk 3 pontot és, arra a 3 pontra alkalmazzuk a formulát.

Elnevezések	Pontszámok	Formulák
Trapézformula	2	$\frac{1}{2}h\,(y_1 + y_2)$
Simpson formula	3	$\frac{1}{3}h\,(y_1 + 4y_2 + y_3)$
Simpson 3/8 formula	4	$\frac{3}{8}h\,(y_1 + 3y_2 + 3y_3 + y_4)$
Boole formula	5	$\frac{2}{45}h\,(7y_1 + 32y_2 + 12y_3 + 32y_4 + 7y_5)$

3.4. Nyitott Newton–Cotes formulák

A nyitott Newton–Cotes formuláknál akár egy pont ismeretével is lehet közelítést adni. A két pontot igénylő nyitott Newton–Cotes formulát fogom levezetni, amit nyitott trapézformulának neveznek.

3. Tétel. *A nyitott trapézformula képlete* $\frac{3}{2} \cdot h \cdot (y_1 + y_2)$*, ahol* $h = x_2 - x_1$*.*

Bizonyítás. Használjuk fel a trapézformulánál kiszámolt függvényt a mostani bizonyításhoz is, mert itt is két pontot használunk a formulában:

$$\int_{x_0}^{x_3} f(x)\,dx = \int_{x_1-h}^{x_1+2\cdot h} L_2(x)\,dx =$$

$$= \frac{1}{2\cdot h}\,(y_2 - y_1) \cdot \left[x^2\right]_{x_1-h}^{x_1+2\cdot h} + \left(y_1 + \frac{x_1}{h}\cdot y_1 - \frac{x_1}{h}\cdot y_2\right) \cdot [x]_{x_1-h}^{x_1+2\cdot h} =$$

$$= \frac{3}{2} \cdot h \cdot (y_1 + y_2)\,.$$

□

Az előbbi levezetéssel megkaptuk a nyitott trapézformulát. Felsorolok még pár nyitott Newton–Cotes formulát, amit már nem fogok levezetni, mert nem fogunk foglalkozni velük a továbbiakban. Az alábbi táblázatban a pontszámok a formulák által használt pontok számát jelölik

Elnevezések	Pontszámok	Formulák
Téglalapszabály	1	$2hy_1$
Nyitott trapézformula	2	$\frac{3}{2}h\left(y_1+y_2\right)$
Milne formula	3	$\frac{4}{3}h\left(2y_1-y_2+2y_3\right)$
Gillenator formula	4	$\frac{5}{24}h\left(11y_1+y_2+y_3+11y_4\right)$

A nyitott Newton–Cotes formulákkal nem foglalkozunk többet, inkább a zárt Newton–Cotes formulák közül a trapéz- és Simpson formulával foglalkozunk.

3.5. Programkóddal kapott eredmények

A programkódok, amiket megemlítek, megtalálhatóak a Függelékben. Az alábbi táblázatban a kod1.m kódot használtuk az eredmények megállapításához. A függvényeket az s2.m programmal generáltuk, a pontos integrálnak az értékét pedig az s3.m kóddal számoltuk ki. A ponthalmaz, amin néztük a formulákat, a trapézformulánál az $x_1 = 0$; $x_2 = 1$ pontok, míg a Simpson formulánál az $x_1 = 0$; $x_2 = 0,5$; $x_3 = 1$ pontok voltak. A [0,1] intervallumon vizsgáljuk a függvényeket. A táblázatban a Trapéz és a Simpson oszlop a közelítés hibáját mutatja, az eredeti függvény integráljához képest.

Függvény	Trapéz	Simpson
$9x^6+7x^4+6x^3+7x^2+9x+3$	22	0,3664
3	0	0
$5x^7+4x^6+2x^5+3x^4+7x^3+8x^2+7x+6$	7,9536	0,4379
$5x^9+5x^8+3x^7+7x^6+4x^5+3x^4+6x^3+8x^2+10x+8$	12,6361	1,1192
$4x^5+2x^4+2x^3+7x^2+10x+6$	3,6000	0,1000
$2x^8+7x^7+3x^6+8x^4+4x^3+x^2+9x+4$	8,0409	0,6138
$2x^5+7x^4+4x^3+5x^2+4x+3$	4,6000	0,1000
$8x^2+2x+10$	1,3333	0
$3x^8+2x^7+3x^6+2x^5+2x^4+4x^3+8x^2+6x+2$	6,5881	0,4292
$10x^7+x^6+9x^5+2x^4+4x^3+3x^2+4x+3$	9,2071	0,7071

A formulák nem adnak elég pontos eredményt ilyen formában. Vegyünk több pontot és használjuk a formulákat a részintervallumokra, ezzel is növelve a formulák pontosságát. Ezeket hívjuk összetett Newton–Cotes formuláknak.

3.6. Összetett Newton–Cotes formulák

6. Definíció. *Az összetett trapézformula a részintervallumokon használt trapézformulának az összessége.*

7. Definíció. *Az összetett Simpson formula a részintervallumokon használt Simpson formulának az összessége.*

A Simpson formulát csak bizonyos ponthalmazra lehet alkalmazni, ahol a pontok ekvidisztánsak és páratlan darab pont van, tehát a trapézformulát is ilyen tulajdonságú ponthalmazon vizsgáljuk.

3.7. Programkóddal kapott eredmények

A programkódok, amiket megemlítek, megtalálhatóak a Függelékben. A következő formulákat a [0,1] intervallumon vizsgáljuk. A kod2.m programot használjuk a formulák és az eredeti integrál különbségének meghatározására. Az integrál pontos értékét az s3.m kóddal kaptuk meg, a függvényt az s2.m programmal generáltuk, míg a pontokat az s4.m programmal generáltuk, hogy a Simpson formula feltételei teljesüljenek a ponthalmazra. A táblázatban a Trapéz és a Simpson oszlop a közelítés hibáját mutatja, az eredeti függvény integráljához képest.

Függvény	Pontszám	Trapéz	Simpson
$4x^3 + 6x^2 + 6x + 5$	107	0,3668	2,3093e-14
$4x^2 + 4x + 1$	93	0,0819	1,7764e-15
$5x^2 + 9x + 2$	103	0,2454	1,7764e-15
$10x^3 + 7x^2 + 7x + 10$	101	0,5201	7,1054e-15
$4x^3 + x^2 + 7x + 7$	107	0,6335	1,4211e-14
$8x^2 + 4x + 3$	99	0,5004	3,5527e-15
$6x^4 + 9x^3 + 2x^2 + 4x + 3$	121	0,4276	2,6026e-08
$10x^5 + 8x^4 + 5x^3 + x^2 + 9x + 3$	125	0,5740	1,8320e-07
$2x^4 + 9x^3 + 3x^2 + 10x + 8$	113	0,3984	1,3733e-08
$5x$	105	0,0058	1,3323e-15

A formulák sokkal pontosabb közelítést adtak, mint ahogy az várható is volt. Nézzük meg, hogy van-e valamilyen összefüggés a részintervallumok száma és a hiba nagysága között.

3.8. Hiba rendszerszerűsége programmal

A programkódok, amiket megemlítek, megtalálhatóak a Függelékben. Az eredményeket a t13.m programmal kaptuk, ahol a függvényt az s2.m programból kaptuk, a pontos integrál értékét pedig az s3.m program segítségével kaptuk. A [0,1] intervallumon vizsgáljuk a függvényt és a részintervallumok számának a kétszerezésével kaptuk ezeket a táblázatokat. A hányados oszlop az eggyel fölötte levő sorban levő trapéz hibája oszlop eredménye osztva a vele egy sorban levő trapéz hibája oszlopban levő eredménnyel jött ki.

Függvény	Részintervallumok	Trapéz hibája	Hányados
$7x^5 + 9x^4 + 8x^3 + 2x^2 + 2$	1	7,3667	-
	2	2,0073	3,6699
	4	0,5122	3,9189
	8	0,1287	3,9797
	16	0,0322	3,9968
	32	0,0081	3,9753
	64	0,0020	4,05
	128	5,0354e-04	3,9725
	256	1,2588e-04	3,9994
	512	3,1471e-05	3,9998
	1024	7,8678e-06	3,9999

Függvény	Részintervallumok	Trapéz hibája	Hányados
$x^6 + 2x^5 + 5x^4 + 5x^3 + x^2 + 5x + 1$	1	3,9405	-
	2	1,0733	3,6713
	4	0,2741	3,915
	8	0,0689	3,978
	16	0,0172	4,0058
	32	0,0043	4
	64	0,0011	3,909
	128	2,6957e-04	4,0805
	256	6,7393e-05	3,9999
	512	1,6848e-05	4
	1024	4,2121e-06	3,9999

A programkódok, amiket megemlítek, megtalálhatóak a Függelékben. Az eredményeket a s12.m programmal kaptuk, ahol a függvényt az s2.m prog-

ramból kaptuk, a pontos integrál értékét pedig az s3.m program segítségével kaptuk. A [0,1] intervallumon vizsgáljuk a függvényt és a részintervallumok számának a kétszerezésével kaptuk ezeket a táblázatokat. A hányados oszlop az eggyel fölötte levő sorban levő Simpson oszlop eredménye osztva a vele egy sorban levő Simpson oszlopban levő eredménnyel jött ki. A táblázatban a Simpson oszlop a közelítés hibáját mutatja, az eredeti függvény integráljához képest.

Függvény	Részintervallumok	Simpson	Hányados
$2x^5+3x^4+7x^3+6x^2+2x+10$	2	0,0667	-
	4	0,0042	15,8809
	8	2,6042e-04	16,1277
	16	1,6276e-05	16,0002
	32	1,0173e-06	15,9992
	64	6,3578e-08	16,0008
	128	3,9736e-09	16,0001
	256	2,4835e-10	16
	512	1,5522e-11	15,9998
	1024	9,7167e-13	15,9745

Függvény	Részintervallumok	Simpson	Hányados
$2x^6+4x^5+3x^4+9x^3+8x^2+6x+9$	2	0,1768	-
	4	0,0117	15,1111
	8	7,4506e-04	15,7034
	16	4,6737e-05	15,9415
	32	2,9237e-06	15,9855
	64	1,8277e-07	15,9966
	128	1,1424e-08	15,9987
	256	7,1403e-10	15,9993
	512	4,4626e-11	16,0003
	1024	2,7924e-12	15,9812

A hányados oszlopból jól kivehető, hogy a trapézformula hibája az intervallumok kétszerezésével a negyedére csökken, míg a Simpson formula hibája az intervallumok kétszerezésével a tizenhatodára csökken. A következő részben bebizonyítjuk, hogy miért jött ki ez az eredmény, a formulák hibájának felső becslésével.

3.9. A hibák képletei

Hivatkozás a következő tételhez és bizonyításához!

4. Tétel. *A trapézformula hibáját jelöljük T_{hiba}-val és legyen ξ az $[a,b]$ intervallumnak az a pontja, ahol a függvényérték maximális. Az intervallumok száma legyen N. Ekkor ezt a becslést írhatjuk fel a trapézformula hibájára:*

$$T_{hiba} \leq \frac{(b-a)^3}{12 \cdot N^2} \cdot \left| f^{(2)}(\xi) \right|.$$

Bizonyítás. Vegyük az eredeti $f(x)$ függvény integrálját egy részintervallumon, amit átírva egy másik alakra képesek leszünk kapni egy olyan egyenletet,ami az $f''(x)$-t magában foglalja:

$$\int_{x_i}^{x_{i+1}} f(x)\,dx = \int_0^h f(t+x_i)\,dt.$$

Használjuk a következő parciális integrálást a továbbiakban:

$$\int_a^b u\,v' = [u \cdot v]_a^b - \int_a^b v\,u'.$$

Használjuk az előbbi formulát $u = f(t+x_i)$ és $v = t + c_1$ helyettesítésekkel, ahol c_1 konstans:

$$\int_0^h f(t+x_i)\,dt = [(t+c_1)\cdot f(t+x_1)]_0^h - \int_0^h (t+c_1)\cdot f'(t+x_i)\,dt.$$

Használjuk a parciális integrálás formuláját $u = f'(t+x_i)$ és $v = \frac{t^2}{2}+c_1\cdot t+c = \frac{(t+c_1)^2-c_1^2}{2} + c = \frac{(t+c_1)^2}{2} + c_2$ helyettesítésekkel, ahol c és c_2 konstans:

$$\int_0^h f(t+x_i)\,dt = [(t+c_1)\cdot f(t+x_1)]_0^h - \left[\left(\frac{(t+c_1)^2}{2}+c_2\right)\cdot f'(t+x_i)\right]_0^h +$$

$$+ \int_0^h \left(\frac{(t+c_1)^2}{2}+c_2\right)\cdot f''(t+x_i)\,dt.$$

A továbbiakban úgy kell megválasztani c_1 és c_2 értékeket, hogy az előző egyenletből a trapézformulát és s T_{hiba}-t kapjuk vissza, az egyenlőség jobb oldalának első részének a trapézformulával kell, hogy megegyezen:

$$[(t+c_1)\cdot f(t+x_1)]_0^h = \frac{h}{2}\cdot(y_i + y_{i+1})$$

$$(h + c_1) \cdot y_{i+1} - c_i \cdot y_i = \frac{h}{2} \cdot (y_i + y_{i+1})$$

$$c_1 = \frac{-h}{2}.$$

Az első deriváltat magában foglaló résznek 0-nak kell lennie:

$$\left[\left(\frac{(t + c_1)^2}{2} + c_2 \right) \cdot f'(t + x_i) \right]_0^h = 0$$

$$\left(\frac{(\frac{h}{2})^2}{2} + c_2 \right) \cdot f'(x_{i+1}) - \left(\frac{(\frac{-h}{2})^2}{2} + c_2 \right) \cdot f'(x_i) = 0$$

$$\Rightarrow \frac{(\frac{h}{2})^2}{2} + c_2 = 0$$

$$c_2 = \frac{-h^2}{8}.$$

A második deriváltat magában foglaló rész lesz a T_{hiba} meghatározásához szükséges rész:

$$T_{hiba}(i) = \int_0^h \left(\frac{(t + c_1)^2}{2} + c_2 \right) \cdot f''(t + x_i)\, dt$$

$$T_{hiba} = \int_0^h \left(\frac{(t - \frac{h}{2})^2}{2} - \frac{h^2}{8} \right) \cdot (f''(t + x_0) + \ldots + f''(t + x_{N-1}))\, dt$$

$$\Rightarrow |T_{hiba}| = \left| \int_0^h \left(\frac{(t - \frac{h}{2})^2}{2} - \frac{h^2}{8} \right) \cdot (f''(t + x_0) + \ldots + f''(t + x_{N-1}))\, dt \right| \leq$$

$$\leq \int_0^h \left(\frac{(t - \frac{h}{2})^2}{2} - \frac{h^2}{8} \right) \cdot N \cdot \left| f^{(2)}(\xi) \right|\, dt.$$

Az egyenlőtlenségben lévő integrál kiszámításával és a kapott eredmény abszolútértékének a visszahelyettesítésével kapunk egy felső becslést a trapézformula hibájára:

$$\int_0^h \left(\frac{(t - \frac{h}{2})^2}{2} - \frac{h^2}{8} \right)\, dt = \left[\frac{(t - \frac{h}{2})^3}{6} - \frac{h^2}{8} \cdot t \right]_0^h =$$

$$= \frac{(\frac{h}{2})^3}{6} - \frac{h^3}{8} - \frac{(\frac{-h}{2})^3}{6} = \frac{h^3 - 6 \cdot h^3 + h^3}{48} = \frac{-h^3}{12}.$$

Az egyenlőtlenségbe való behelyetesítéssel kapunk egy felső becslést a trapézformula hibájára:

$$T_{hiba} \leq N \cdot \left|f^{(2)}(\xi)\right| \cdot \frac{h^3}{12} = \frac{N \cdot \left|f^{(2)}(\xi)\right| \cdot (\frac{b-a}{N})^3}{12} = \frac{(b-a)^3}{12 \cdot N^2} \cdot \left|f^{(2)}(\xi)\right|.$$

Visszakaptuk az egyenlőtlenséget, tehát bebizonyítottuk a tételt. □

Az $\frac{1}{N^2}$ szorzó miatt láttuk azt, hogy a pontok kétszerezésénél a trapézformula hibája negyedelődött. A Simpson formulánál tehát a hibaképletben szerepelnie kell egy $\frac{1}{N^4}$ szorzónak.

Hivatkozás a következő tételhez és bizonyításához!

5. Tétel. *A Simpson formula hibáját jelöljük S_{hiba}-val és legyen ξ az $[a, b]$ intervallumnak az a pontja, ahol a függvényérték maximális. A részintervallumok száma legyen N. Ekkor ezt a becslést írhatjuk fel a Simpson formula hibájára.*

$$S_{hiba} \leq \frac{(b-a)^5}{180 \cdot N^4} \cdot \left|f^{(4)}(\xi)\right|$$

Bizonyítás. Definiáljunk egy olyan szorzat integrálját, ahol $f(x)$ 4-szer deriválható és $p(x) = x^4 + a_3x^3 + a_2x^2 + a_1x + a_0$, ahol $a_0, a_1, a_2, a_3 \in \mathbb{R}$

$$\int_a^b f^{(4)}(x)p(x)\,dx.$$

A trapézformula hibájának a bizonyításánál használt módszert fogjuk itt is alkalmazni:

$$\int_a^b uv' = [u \cdot v]_a^b - \int_a^b vu'.$$

Használjuk az $u = p(x)$ és $v = f'''(x)$ helyettesítést:

$$\int_a^b f^{(4)}(x)p(x)\,dx = [p(x)f'''(x)]_a^b - \int_a^b f'''(x)p'(x)\,dx.$$

Legyen $u = p'(x)$ és $v = f''(x)$:

$$\int_a^b f'''(x)p'(x)\,dx = [p'(x)f''(x)]_a^b - \int_a^b f''(x)p''(x)\,dx.$$

Legyen $u = p''(x)$ és $v = f'(x)$:

$$\int_a^b f''(x)p''(x)\,dx = [p''(x)f'(x)]_a^b - \int_a^b f'(x)p'''(x)\,dx.$$

Még mielőtt elvégezzük az utolsó helyettesítést, nézzük meg $p(x)$ negyedik deriváltjának az értékét.

$$p(x) = x^4 + a_3x^3 + a_2x^2 + a_1x + a_0$$

$$p'(x) = 4x^3 + 3a_3x^2 + 2a_2x + a_1$$

$$p''(x) = 12x^2 + 6a_3x + 2a_2$$

$$p'''(x) = 24x + 6a_3$$

$$p^{(4)} = 24.$$

Az utolsó behelyettesítésnél $u = p'''(x)$ és $v = f(x)$

$$\int_a^b f'(x)p'''(x)\,dx = [p'''(x)f(x)]_a^b - 24\int_a^b f(x)\,dx.$$

Írjuk fel az eredeti integrált:

$$\int_a^b f^{(4)}(x)p(x)\,dx = [p(x)f'''(x)]_a^b - [p'(x)f''(x)]_a^b +$$

$$+[p''(x)f'(x)]_a^b - [p'''(x)f(x)]_a^b + 24\int_a^b f(x)\,dx.$$

Azaz $c = \frac{b-a}{2}$ és jelölje c^- a bal oldali közelítést c^+ pedig a jobb oldali közelítést.
Bontsuk ki ezeknek az ismeretében az előző egyenlet jobb oldalát,

$$\int_a^b f^{(4)}(x)p(x)\,dx = -f'''(a)p(a) + f'''(c)(p(c^-) - p(c^+)) + f'''(b)p(b) +$$

$$+f''(a)p'(a) - f''(c)(p'(c^-) - p'(c^+)) - f''(b)p'(b) - f'(a)p''(a) + f'(c)(p''(c^-) - p''(c^+)) +$$

$$+f'(b)p''(b) + f(a)p'''(a) - f(c)(p'''(c^-) - p'''(c^+)) - f(b)p'''(b) + 24\int_a^b f(x)\,dx.$$

A hiba korlátozása miatt:

$$0 = p(a) = p'(a) = p''(a) = p''(b) = p'(b) = p(b).$$

A továbbiakban a $p'''(x)$-t fogjuk kiszámolni, hogy a hibát tartalmazó egyenlőségben az ismeretlen tagokat ismertté tegyük.

$$p(x) = (x-a)^3(x+d_1) \text{ ha } a \leq x \leq c$$

$$p(x) = (x-b)^3(x+d_2) \text{ ha } c \leq x \leq b$$

Mivel $p(x)$ folytonos függvény, ezért c behelyettesítésekor a két egyenletnek meg kell egyeznie:

$$(c-a)^3(c+d_1) = (c-b)^3(c+d_2) = -(c-b)^3(c+d_1)$$

$$d_1 + d_2 = -2c = -2\left(\frac{a+b}{2}\right) = -(a+b).$$

$$p'(x) = (x-a)^2(4x+3d_1-a) \text{ ha } a \leq x \leq c$$

$$p'(x) = (x-b)^2(4x+3d_2-b) \text{ ha } c \leq x \leq b$$

A $p'(x)$ folytonos c-ben, tehát behelyetesítéskor a két egyenletnek meg kell egyeznie.

$$4c + 3d_1 - a = 4c + 3d_2 - b$$

$$b - a = 3(d_2 - d_1)$$

A két egyenlet ismeretében képesek vagyunk meghatározni d_1 és d_2 értékeket:

$$d_1 = -d_2 - a - b$$

$$\frac{b-a}{3} = d_2 + d_2 + a + b$$

$$d_2 = \frac{-(2a+b)}{3} \Rightarrow d_1 = \frac{-(a+2b)}{3}.$$

A d_1 és d_2 behelyettesítésével határozzuk meg a $p''(x)$ függvényeket:

$$p''(x) = 4(x-a)(3x-2a-b), \text{ ha } a \leq x \leq c$$

$$p''(x) = 4(x-b)(3x-2b-a), \text{ ha } c \leq x \leq b$$

Helyettesítsük be a c^--t és a c^+-t a megfelelő egyenletbe, és nézzük meg mi jön ki:

$$p''(c^-) = 4(c-a)(3c-2a-b) = 4\frac{(b-a)}{2}\frac{(b-a)}{2} = (b-a)^2$$

$$p''(c^+) = 4(c-b)(3c-a-2b) = \frac{(a-b)}{2}\frac{(a-b)}{2} = (a-b)^2 = (b-a)^2.$$

Azt kaptuk, hogy $p(x)$ folytonos a második deriváltjában is. Vizsgáljuk meg a harmadik deriváltakat, az a, c^-, c^+, b behelyettesítésekkel.

$$p'''(x) = 4(6x-5a-b) \text{ ha } a \le x \le c$$

$$p'''(x) = 4(6x-a-5b) \text{ ha } c \le x \le b$$

$$p'''(a) = 4(a-b)$$

$$p'''(c^-) = 2(4b-4a) = 8(b-a)$$

$$p'''(c^+) = 2(4a-4b) = 8(a-b)$$

$$p'''(b) = 4(b-a).$$

Az alábbi egyenletbe behelyettesítjük azokat az eredményeket, amiket most kaptunk:

$$\int_a^b f^{(4)}(x)p(x)\,dx = f(a)p'''(a) - f(c)(p'''(c^-) - p'''(c^+)) - f(b)p'''(b) + 24\int_a^b f(x)\,dx$$

$$\int_a^b f^{(4)}(x)p(x)\,dx = 4(a-b)f(a) - 16(b-a)f(c) - 4(b-a)f(b) + 24\int_a^b f(x)\,dx$$

$$\int_a^b f^{(4)}(x)p(x)\,dx = -4(b-a)(f(a)+4f(b)+f(c)) + 24\int_a^b f(x)\,dx$$

$$\int_a^b f(x)\,dx = \frac{(b-a)}{6}(f(a)+4f(b)+f(c)) + \frac{1}{24}\int_a^b f^{(4)}(x)p(x)\,dx.$$

Kaptunk egy integrált, aminek a megoldása ad egy felső becslést a Simpson formula hibájára egy részintervallumon. Jelöljük S_{rhiba}-val a Simpson formula részintervallumon vett hibáját:

$$S_{rhiba} = \frac{1}{24}\int_a^b f^{(4)}(x)p(x)\,dx \le \frac{f^{(4)}(\xi)}{24}\int_a^b |p(x)|\,dx$$

$$\int_a^b |p(x)|\,dx = \int_a^c \left|(x-a)^3\left(x-\frac{a}{3}-\frac{2b}{3}\right)\right|\,dx + \int_c^b \left|(x-b)^3\left(x-\frac{2a}{3}-\frac{b}{3}\right)\right|\,dx$$

$$\int_a^b p(x)\,dx = \int_a^c (x-a)^3\left(\frac{a}{3}+\frac{2b}{3}-x\right)\,dx + \int_c^b (b-x)^3\left(x-\frac{2a}{3}-\frac{b}{3}\right)\,dx.$$

A jobb oldali integrálra alkalmazzunk helyettesítési integrált, ahol $t = a + b - x$:

$$\int_a^b p(x)\,dx = \int_a^c (x-a)^3\left(\frac{a}{3}+\frac{2b}{3}-x\right)\,dx - \int_c^a (t-a)^3\left(\frac{a}{3}+\frac{2b}{3}-t\right)\,dt$$

$$\int_a^b p(x)\,dx = 2\int_a^c (x-a)^3\left(\frac{a}{3}+\frac{2b}{3}-x\right)\,dx$$

$$\int_a^b p(x)\,dx = 2\int_a^c (x-a)^3\left(\frac{2(b-a)}{3}-(x-a)\right)\,dx =$$

$$= \frac{4(b-a)}{3}\int_a^c (x-a)^3 - 2\int_a^c (x-a)^4\,dx =$$

$$= \frac{4}{3}(b-a)\left[\frac{(x-a)^4}{4}\right]_a^c - 2\left[\frac{(x-a)^5}{5}\right]_a^c =$$

$$= \frac{(b-a)^5}{48} - \frac{(b-a)^5}{80} = \frac{2(b-a)^5}{240} = \frac{(b-a)^5}{120}.$$

Helyettesítsük vissza a kapott eredmény az S_{rhiba} egyenlőtlenségbe.

$$S_{rhiba} \le \frac{f^{(4)}(\xi)}{24}\frac{(b-a)^5}{120} = \frac{f^{(4)}(\xi)(b-a)^5}{2880}.$$

Definiáljunk egy $p(x)$-hez hasonló funkciót aminek a szummázásával megkapjuk a keresett felső becslést a Simpson formula hibájára.

$$P(x) = (x-x_{2i-2})^3\left(x-\frac{x_{2i-2}}{3}-\frac{2x_{2i}}{3}\right), \text{ ahol } x_{2i-2} \le x \le x_{2i-1}$$

$$P(x) = (x-x_{2i})^3\left(x-\frac{2x_{2i-2}}{3}-\frac{x_{2i}}{3}\right), \text{ ahol } x_{2i-1} \le x \le x_{2i}$$

$$S_{hiba} \le \frac{f^{(4)}(\xi)}{12}\sum_{i=1}^{\frac{n}{2}}\int_{x_{2i-2}}^{x_{2i-1}} P(x)\,dx.$$

Számoljuk ki az integrált a hiba felső becslésének meghatározásához.

$$\int_{x_{2i-2}}^{x_{2i-1}} P(x)\,dx = \int_0^h t^3 \left(\frac{x_{2i-2}}{3} + \frac{2x_{2i}}{3} - t - \frac{3x_{2i-2}}{3} \right) dt =$$

$$= \int_0^h t^3 \left(\frac{4}{3}h - t \right) = \frac{4}{3}h \int_0^h t^3\,dt - \int_0^h t^t\,dt =$$

$$= \frac{h^5}{3} - \frac{h^3}{5} = \frac{2h^5}{15}.$$

Helyettesítsünk vissza az S_{hiba} egyenlőtlenségbe:

$$S_{hiba} \leq \frac{f^{(4)}(\xi)}{12} \sum_{i=1}^{\frac{n}{2}} \frac{2h^5}{15} =$$

$$= \frac{f^{(4)}(\xi)}{12} \frac{n}{2} \frac{2h^5}{15} = \frac{f^{(4)}(\xi) n}{180} h^5$$

$$h = \frac{b-a}{n}$$

$$S_{hiba} \leq \frac{(b-a)^5}{180n^4} f^{(4)}(\xi),$$

tehát bebizonyítottuk a tételt. □

4. Spline interpoláció

4.1. Bevezetés

Jelen fejezetben felhasználjuk az előző két fejezetben látott vizsgálati módszereket. Tehát a Lagrange-interpolációhoz hasonlóan itt is szükségünk lesz az integrálásra, és a Newton–Cotes formulákhoz hasonlóan itt is a részintervallumokon kapott eredmények összesége lesz az az eredmény, ami közelíteni fogja az eredeti függvény integráljának az értékét.

4.2. Alapgondolat

Szeretnénk egy olyan módszert, amivel egy részintervallumon interpolálunk egy magasabb fokú polinomot, a pontosság növelésének érdekében. Ezeket a módszereket hívjuk spline interpolációnak. A spline interpolációkat a gyakorlatban leginkább harmadfokig használják, amit köbös spline-nak nevezünk, tehát én is eddig fogok velük foglalkozni. A köbös spline használatához további feltételek is szükségesek.

4.3. Elsőfokú spline

Az elsőfokú spline interpolációnál elsőfokú polinomokat, tehát szakaszokat kapunk a részintervallumokra. Foglalkoztunk már ilyennel a Newton–Cotes formuláknál, ezt neveztük trapézformulának:

$$\frac{1}{2} \cdot h \cdot (y_1 + y_2) = \int_a^b f(x)\,dx.$$

Az összetett Newton–Cotes formulák hibájának a részénél már láttuk, hogy a módszer csak sok függvényérték ismeretében ad elég pontos megoldást, de cserébe nem kellenek újabb ismeretek a formula meghatározásához. A továbbiakban két példán keresztül mutatok új módszereket az integrálközelítő formula meghatározásához.

4.4. Osztott differencia táblázat

A másodfokú és a harmadfokú spline alkalmazásához szükségünk van az osztott differencia táblázatokra. Az alábbiakban az x_i pontokhoz tartozó y_i függvényértékeket felhasználva kapunk egy olyan táblázatot, ahol a következő

jelöléseket használjuk:

$$f[y_i, y_{i+1}] = \frac{y_{i+1} - y_i}{x_{i+1} - x_i}$$

$$f[y_i, y_{i+1}, y_{i+2}] = \frac{f[y_{i+1}, y_{i+2}] - f[y_i, y_{i+1}]}{x_{i+2} - x_i}.$$

z_1				
z_2	$f[z_1, z_2]$			
z_3	$f[z_2, z_3]$	$f[z_1, z_2, z_3]$		
...			...	
z_n	...	...	...	$f[z_1, ..., z_n]$

Ennek a táblázatnak a lépcső elemeit véve kapunk egy $n-1$-ed fokú polinomot, ami így néz ki:

$$p(x) = y_1 + (x - y_1) \cdot f[y_1, y_2] + (x - y_1) \cdot (x - y_2) \cdot f[y_1, y_2, y_3] + \dots$$

A spline interpolációhoz elég lesz a legfeljebb harmadrendű osztott differencia, mivel a köbös splinenál harmadfokú polinomokat szeretnénk kapni.

4.5. Másodfokú spline

A másodfokú spline interpolációnál másodfokú polinomokat kapunk az $[x_i, x_{i+1}]$ intervallumokra. A másodrendű spline-t a másodrendű osztott differencia táblázat alapján lehet kiszámolni, ahol a következő helyetesítésekkel dolgozunk:

$$z_1 = x_i, z_2 = x_i, z_3 = x_{i+1}$$

A táblázatba való behelyettesítés után a következő $p_i(x)$ polinomot kapjuk:

$$p_i(x) = y_i + f[x_i, x_i] \cdot (x - x_i) + f[x_i, x_i, x_{i+1}] \cdot (x - x_i)^2$$

A behelyettesítés után megkapjuk a keresett másodfokú polinomot a részintervallumon.
Megnézzük, hány feltételre van szükségünk, és hogy hány feltételünk adott a másodfokú spline megtalálásához.
Mivel N részintervallumunk adott, mindegyik részintervallumon egy másodfokú polinomunk van, tehát:

$$p_i(x) = A \cdot x^2 + B \cdot x + C \;\text{ ahol}, i = 1, \dots, N, \;\; A, B, C \in \mathbb{R}.$$

Ebből adódik, hogy $3N$ feltételünk van. Most nézzük meg az adott feltételek számát. Minden részintervallumon vett másodfokú polinomunk végpontjaiban meg kell egyeznie a függvényértékkel, tehát $2N$ feltétel. A köztes pontokban találkozó polinomoknak az első deriváltjának is meg kell egyezniük, ami $N-1$ feltételt jelent, tehát összesen $3N-1$ feltételt kapunk. A másodfokú spline nem egyértelmű függvény, mert $3N$ feltétel kell, de $3N-1$ adott feltételünk van. A további egy feltétel kétféleképpen állhat elő. Vagy a részintervallum bal oldali végpontjának a deriváltértékének a megadásával, vagy a részintervallum jobb oldali végpontjának a deriváltértékének a megadásával.

4.6. Hermite-interpoláció

A köbös spline levezetéséhez kell a Hermite-interpoláció. A Hermite-interpoláció alkalmazásához minden $[x_i, x_{i+1}]$ részintervallumban az alábbi értékeket kell ismernünk: részintervallum végeiben lévő függvényértékeket, valamint az első deriváltjukat:

$$x_i; y_i = f(x_i); y_i^{'} = f^{'}(x_i); x_{i+1}; y_{i+1} = f(x_{i+1}); y_{i+1}^{'} = f^{'}(x_{i+1}),$$

ahol $i = 1, \ldots, N$.
A harmadrendű osztott differenciánál, a táblázatba való behelyettesítéssel a következőképpen számoljuk ki a polinomot:

$$z_1 = x_i, z_2 = x_i, z_3 = x_{i+1}, z_4 = x_{i+1}.$$

A behelyettesítés után a következő képletet kapjuk a $p_i(x)$ polinomra. Nevezzük el az $f[x_i, x_{i+1}]$ értéket δ_i-nek az egyszerűség kedvéért:

$$p_i(x) = y_i + (x - x_i) \cdot \delta_i \cdot y_i + (x - x_i)^2 \cdot \delta_i^2 \cdot y_i + (x - x_i)^2 \cdot (x - x_{i+1}) \cdot \delta_i^3 \cdot y_i.$$

Így kapjuk meg a keresett harmadfokú polinomot a megadott részitervallumon.

4.7. Harmadfokú spline

A harmadfokú spline-t szokták a leggyakrabban használni, mert elég pontos eredményt ad, és kevesebb új feltételre van szükség a függvény megállapításához.

Olyan $s(x)$ függvényt keresünk az adott $[a,b]$ intervallumon, ahol adottak az (x_i, y_i) értékek. Legyen $s : [a,b] \to \mathbb{R}$ az alábbi tulajdonságokkal.

$$s(x_i) = y_i,\ i = 0, 1, \ldots, n$$

$$s \in C^2[a,b]$$

$$\int_a^b (s''(x))^2\,dx \text{ minimális}$$

Ez egy variációs probléma, melynek megoldásához a következőt kell kiszámolnunk:

$$f(x) = s(x) + \epsilon h(x),\ \epsilon \in \mathbb{R},\ h \in C^2[a,b],\ h(x_i) = 0$$

$$\int_a^b (s''(x))^2\,dx \le \int_a^b (s''(x) + \epsilon \cdot h''(x))^2\,dx =$$

$$= \int_a^b (s''(x))^2\,dx + 2 \cdot \epsilon \cdot \int_a^b s''(x) \cdot h''(x)\,dx + \epsilon^2 \cdot \int_a^b (h''(x))^2\,dx$$

$$0 \le 2 \cdot \epsilon \cdot \int_a^b s''(x) \cdot h''(x)\,dx + \epsilon^2 \cdot \int_a^b (h''(x))^2\,dx$$

$$0 \le \epsilon^2 \cdot \int_a^b (h''(x))^2\,dx$$

$$\Rightarrow 2 \cdot \epsilon \cdot \int_a^b s''(x) \cdot h''(x)\,dx = 0 \text{ kell}$$

$$\int_a^b s''(x) \cdot h''(x)\,dx = s''(b) \cdot h'(b) - s''(a) \cdot h'(a) - \int_a^b s'''(x) \cdot h'(x)\,dx.$$

A kapott képletben szereplő kifejezésnek kell 0-nak lennie, ezért legyen az egyenlőség mindkét oldala 0:

$$s''(b) \cdot h'(b) - s''(a) \cdot h'(a) = 0$$

$$\int_a^b s'''(x) \cdot h'(x)\,dx = 0.$$

A kapott egyenlet akkor teljesül, ha az s háromszor differenciálható a részintervallumon.

8. Definíció. *Legyenek x_i-k az alappontok. Egy $s \in C^2[a,b]$ függvényt köbös spline-nak nevezünk, ha $[x_i, x_{i+1}]$ intervallumon egy harmadfokú polinomot alkot és azok összeérnek az x_j csatlakozási pontokban, ahol $j = 2, \ldots, n-1$.*

A feltételek számának megadásának menete hasonló a másodfokú splinenél látottakhoz. Mivel N részintervallumunk van és mindegyik részintervallumon egy harmadfokú polinomunk, tehát:

$$p_i(x) = A \cdot x^3 + B \cdot x^2 + C \cdot x + D, \text{ ahol } i = 1, \ldots, N.$$

Ebből adódik, hogy $4N$ feltételünk van, most nézzük meg az adott feltételek számát. Minden részintervallumon vett harmadfokú polinomunk végpontjaiban meg kell egyeznie a függvényértékekkel, tehát ez $2N$ feltétel. A köztes pontokban találkozó polinomoknak az első és a második deriváltjainak az értékének is meg kell egyezniük, ami $2N-2$ feltételt jelent, tehát összesen $4N-2$ adott feltételünk lesz.
A másodfokú spline-hoz hasonlóan itt sincs minden feltétel megadva. A köbös spline-nál $4N$ feltétel kell, de csak $4N-2$ feltétel adott. A további két feltétel megválasztása befolyásolja a spline függvényt, egy feltétel különféle megoldásaival három különböző spline függvényt kaphatunk.
Az alábbi feltételnek kell teljesülnie s-re:

$$s''(b) \cdot \left(f'(b) - s'(b) \right) = s''(a) \cdot \left(f'(a) - s'(a) \right).$$

Ez többféleképpen is teljesülhet:
Természetes spline:

$$s''(a) = s''(b) = 0.$$

Megfeszített spline:

$$s'(a) = P\,, s'(b) = Q \text{ adottak.}$$

Periodikus spline:

$$ha\ f'(a) = f'(b),\ akkor\ s'(a) = s'(b),\ s''(a) = s''(b).$$

Hogyan kapjuk meg a harmadfokú függvényt a spline interpolációhoz az alappontok, a hozzájuk tartozó függvényértéket és a hozzájuk tartozó első derivált értékei alapján? Az előző részben vezettük be a Hermite-interpolációt, ami pont ezekkel az adatokkal dolgozik, és egy harmadfokú polinomot ad

vissza eredményként. A Hermite-interpoláció meghatározza egy részintervallum harmadfokú polinomját a harmadrendű differencia táblázat segítségével, a következő képlet adja meg egy részintervallum harmadfokú polinomját a Hermite-interpoláció részben bevezetett jelölésekkel:

$$p_i(x) = y_i + (x - x_i) \cdot \delta_i \cdot y_i + (x - x_i)^2 \cdot \delta_i^2 \cdot y_i + (x - x_i)^2 \cdot (x - x_{i+1}) \cdot \delta_i^3 \cdot y_i,$$

ahol $i = 1, \ldots, N$.
A részintervallumok összessége megadja a harmadfokú spline-t, ami egy folytonos függvény, de nem egy polinom.

5. A programok összehasonlítása

A programkódok, amiket megemlítek, megtalálhatóak a Függelékben. A következő formulákat a [0,1] intervallumon vizsgáljuk. A kod3.m programot használjuk a formulák és az eredeti integrál különbségének meghatározására. Az integrál pontos értékét az s3.m kóddal kaptuk meg, a függvényt az s2.m programmal generáltuk, míg a pontokat az s1.m programmal generáltuk. Az alsó táblázatban az oszlopok a közelítés hibáját mutatja, az eredeti függvény integráljához képest.

Függvény	Pontszám
$9x^7 + x^6 + 9x^5 + 6x^4 + x^3 + 3x^2 + 5x + 10$	965
$2x^8 + 9x^7 + 8x^6 + 6x^5 + 4x^4 + 3x^3 + 8x^2 + 2x + 1$	767
$8x^4 + 4x^3 + x^2 + 6x + 9$	194
$3x^2 + 5x + 3$	832
$x^8 + 4x^7 + 4x^6 + 7x^5 + 6x^4 + 8x^3 + 4x^2 + 2x + 1$	772
$9x^7 + x^6 + 9x^5 + 6x^4 + x^3 + 3x^2 + 5x + 10$	965
$2x^8 + 9x^7 + 8x^6 + 6x^5 + 4x^4 + 3x^3 + 8x^2 + 2x + 1$	767
$8x^4 + 4x^3 + x^2 + 6x + 9$	194
$3x^2 + 5x + 3$	832
$9x^7 + x^6 + 9x^5 + 6x^4 + x^3 + 3x^2 + 5x + 10$	965

A táblázatból látszik, hogy a Lagrange-interpolációval kapott eredmények közel sem olyan jók, mint a trapézformula vagy a köbös spline eredményei.

Köbös Spline	Trapéz	Lagrange 4 pontra	Lagrange 5 pontra
0	0,5891	3,6464e+03	4,7260e+03
0	0,1512	171,6447	2,1563e+08
0	0,2857	905,9297	702,9828
0	0,0780	1,7940e+03	1,6387e+11
0	0,2109	845,2767	3,3601e+11
0	0,5891	3,6464e+03	4,7260e+03
0	0,1512	171,6447	2,1563e+08
0	0,2857	905,9297	702,9828
0	0,0780	1,7940e+03	1,6387e+11
0	0,5891	3,6464e+03	4,7260e+03

A programkódok, amiket megemlítek, megtalálhatóak a Függelékben. A következő formulákat a [0,1] intervallumon vizsgáljuk. A kod4.m programot használjuk a formulák és az eredeti integrál különbségének meghatározására. Az integrál pontos értékét az s3.m kóddal kaptuk meg, a függvényt az s2.m programmal generáltuk, míg a pontokat az s4.m programmal generáltuk, hogy a Simpson formulát is tudjuk alkalmazni a mostani összehasonlításban. Az alsó táblázatban az oszlopok a közelítés hibáját mutatja, az eredeti függvény integráljához képest.

Függvény	Pontszám
$2x^8 + 9x^7 + 8x^6 + 6x^5 + 4x^4 + 3x^3 + 8x^2 + 2x + 1$	97
$8x^8 + 7x^7 + 8x^6 + 4x^5 + 6x^4 + 6x^3 + 5x^2 + 3x + 2$	95
$4x^9 + 10x^8 + 3x^7 + 9x^6 + 5x^5 + 4x^4 + 2x^3 + x^2 + 3x + 7$	103
$3x^5 + 9x^4 + x^3 + 7x^2 + 6x + 8$	95
$9x^7 + x^6 + 9x^5 + 6x^4 + x^3 + 3x^2 + 5x + 10$	95
$3x^9 + 6x^8 + 2x^7 + 8x^6 + 3x^5 + 5x^4 + 7x^3 + 9x^2 + 10x + 5$	107
10	115
$5x^4 + 3x^3 + 7x^2 + 2x + 7$	103
$3x^2 + 4x + 5$	97
$6x^7 + 2x^6 + 2x^5 + 9x^4 + 5x^2 + 2x + 10$	107

A Simpson formula jobb közelítést ad, mint a trapézformula, de a köbös spline minden vizsgált esetben pontos közelítést adott a függvények [0,1] intervallumon vett integráljának az értékére.

Köbös Spline	Trapéz	Lagrange 4 pontra	Lagrange 5 pontra	Simpson
0	1,3070	75,6439	343,1431	9,4457e-07
0	0,4542	109,5625	2,2702e+06	1,1275e-06
0	0,7406	67,5764	5,3396e+06	2,9412e-06
0	0,5004	443,0001	2,1435e+07	1,0146e-07
0	1,5052	269,9082	4,7025e+08	6,0914e-07
0	0,2829	223,3848	3,2418e+06	1,6978e-06
0	5,3291e-15	370,0000	2,4612e+05	0
0	0,5919	386,5389	7,7260e+12	3,0712e-08
0	0,0396	248,0000	184,0000	0
0	0,0478	383,1945	9,4085e+06	5,6304e-07

6. Alkalmazási területek

Általában a tudósok, fotósok, mérnökök vagy matematikusok használják az interpolációt, hogy megkönnyítse a feladatuk elvégzését.
A tudósok és mérnökök általi mérések nem tudnak folytonosak lenni, ezért az interpoláció segítségével összekötik a mérési eredményeket. A mérési eredmények géppel történő kiértékelésénél fennállhat az a probléma, hogy a kiértékelés nem fog emberi időn belül lefutni, ilyenkor használják az interpolációs formulákat a gyors lefutás érdekében, de cserébe az eredmény nem feltétlenül lesz pontos. A leghétköznapibb dolog, ahol a segítségünkre van az interpoláció, képeknek a nagyítása, elforgatása, amit minden ember használ. A képek nagyításánál a meglévő adatokból kell kitalálnia az interpolációnak a nagyított képet, ezért fordul néha elő, hogy pixeles lesz a nagyított kép, mivel az interpolációval nem sikerült kiszámolni a nagyított kép hiányzó adatait.

7. Összefoglalás

A programok általi eredményeket nézve a köbös spline bizonyult a legjobbnak, ami minden vizsgált esetben igen pontosan viszaadta a függvény integráljának az értékét, de cserébe ismernünk kell a csatlakozási pontokban lévő első deriváltaknak az értékét, amit az esetek többségében nem ismerünk. A Simpson formula jobb eredményeket adott, mint a másik zárt Newton–Cotes beli társa, a trapézformula. A két vizsgált zárt Newton–Cotes formula egész jól közelítette a vizsgált esetekben a függvény integráljának az értékét.

A Lagrange-interpolációval kapott eredmények nem tekinthetőek megfelelő eredményeknek az általunk vizsgált esetekben, biztos léteznek olyan esetek, ahol a Lagrange-interpolációval kapott integrál közelítés az elvárásnak megfelelő hibahatáron belül van, de erre nem láttunk példát.
Minden interpoláló formulának megvan a maga helye, ahol érdemes használni. Ha tudunk kapcsolódási pontokoban első deriváltat számolni, akkor a köbös spline a legjobb megoldás. Ha ekvidisztáns ponthalmazunk van, akkor a Simpson formula jobb választás, mint a trapézformula. A többi esetben a trapézformula jól működik, mert nem igényel a ponthalmazon kívül semmilyen plusz feltételt.

Hivatkozások

[1] HAVASI ÁGNES: Alkalmazott Analízis, egyetemi előadó, jegyzet, Eötvös Loránd Tudományegyetem, 2021

[2] FARAGÓ ISTVÁN, HORVÁTH RÓBERT: Numerikus módszerek, Typotex Kiadó, 2016

[3] Simpson formula hibája

[4] Trapézformula hibája

[5] Newton–Cotes formulák

[6] Alkalmazási területek

Függelék

```
    function fn=kod5()

x=s2();

pont=s1();

disp(l12(1,x,pont));

disp(l12(2,x,pont));

disp(l12(3,x,pont));

disp(l12(4,x,pont));

y=poly2sym(x);

disp(length(pont));

disp(y);


end

    function fn=s3(x)

    y=length(x);

    N=x;

    N(y+1)=0;

    for i=1:y

        N(i)=N(i)/(y-i+1);

    end
```

```
    z=poly2sym(N);

    f=inline(z);

    fn=f(1)-f(0);

end

    function fn=l12(w,x,pont)

    % Lagrange-interpoláció
    % "véletlenszerű" pontok
    % "véletlenszerű" függvény

    % x=s2();

    y=poly2sym(x);

    f=inline(y);

    c=s3(x);

    % pont=s1();

    ered=0;


    for i=1:length(pont)

        s(i)=f(pont(i));

    end

    i=1;

   while i<=length(pont)-w
```

```
for p=i:i+w

I=[1];
K=1;

for j=i:i+w

    if pont(p)~=pont(j)

        T(1)=1;
        T(2)=pont(j)*(-1);

        K=K*(pont(p)-pont(j));
        I=conv(I,T);

    end

end

H=s(p)/K;
M=I*H;

O=s3(M);

ered=ered+O;

end

i=i+w;

if(i+w>length(pont)-w)

    q=length(pont)-w-i;

    for j=1:q

        pont(length(pont)+j)=1;
```

```
                s(length(pont)+j)=f(1);
            end
        end
    end
    fn=abs(ered-c);

end
    function fn=kod4()
    x=s2();
    pont=s4();
    disp(t12(x,pont));
    disp(s13(x,pont));
    disp(cs(x,pont));
    disp(l12(3,x,pont));
    disp(l12(4,x,pont));
    y=poly2sym(x);
    disp(length(pont));
    disp(y);
end
```

```
    function fn=kod3()
    x=s2();
    pont=s1();
    disp(cs(x,pont));
    disp(t12(x,pont));
    disp(l12(3,x,pont));
    disp(l12(4,x,pont));
    y=poly2sym(x);
    disp(length(pont));
    disp(y);

end
    function fn=kod2()
x=s2();
pont=s4();
disp(t12(x,pont));
disp(s13(x,pont));
y=poly2sym(x);
disp(length(pont))
disp(y);
```

```
end

    function fn=kod1()

x=s2();

pont1=[0 1];

pont2=[0 0.5 1];

disp(t12(x,pont1));

disp(s13(x,pont2));

y=poly2sym(x);

disp(y);

end

    function fn=s5(x,a,b)

    y=length(x);

    N=x;

    N(y+1)=0;

    for i=1:y

        N(i)=N(i)/(y-i+1);

    end

    z=poly2sym(N);
```

```
    f=inline(z);

    fn=f(b)-f(a);

end

    function fn=cs(x,pont)


        % x=s2();

        y=poly2sym(x);

        f=inline(y);

        c=s3(x);

        % pont=s1();

        N=length(pont)-1;

        M=zeros(4*N);


        for i=1:N

            M((2*i)-1,4*(i-1)+1)=pont(i)^3;
            M((2*i)-1,4*(i-1)+2)=pont(i)^2;
            M((2*i)-1,4*(i-1)+3)=pont(i)^1;
            M((2*i)-1,4*(i-1)+4)=1;
            K((2*i)-1,1)=f(pont(i));

        end

        for i=1:N

            M((2*i),4*(i-1)+1)=pont(i+1)^3;
            M((2*i),4*(i-1)+2)=pont(i+1)^2;
```

```
    M((2*i),4*(i-1)+3)=pont(i+1)^1;
    M((2*i),4*(i-1)+4)=1;
    K((2*i),1)=f(pont(i+1));

end


for i=2:N


    M(2*N-1+i,4*(i-2)+1)=3*pont(i)^2;
    M(2*N-1+i,4*(i-2)+2)=2*pont(i);
    M(2*N-1+i,4*(i-2)+3)=1;
    M(2*N-1+i,4*(i-2)+5)=(-3)*pont(i)^2;
    M(2*N-1+i,4*(i-2)+6)=(-2)*pont(i);
    M(2*N-1+i,4*(i-2)+7)=(-1);
    M(3*N-2+i,4*(i-2)+1)=6*pont(i);
    M(3*N-2+i,4*(i-2)+2)=2;
    M(3*N-2+i,4*(i-2)+5)=(-6)*pont(i);
    M(3*N-2+i,4*(i-2)+6)=(-2);
    K(2*N-1+i,1)=0;
    K(3*N-2+i,1)=0;

end

%% +2 feltétel

M(4*N-1,1)=6*pont(1);
M(4*N-1,2)=2;
M(4*N,4*N-3)=6*pont(N+1);
M(4*N,4*N-2)=2;
K(4*N-1,1)=0;
K(4*N,1)=0;


S=M\K;

osszeg=0;
```

```
        for i=1:N

            A(1)=S(4*(i-1)+1,1);
            A(2)=S(4*(i-1)+2,1);
            A(3)=S(4*(i-1)+3,1);
            A(4)=S(4*(i-1)+4,1);

            B=s5(x,pont(i),pont(i+1));

            osszeg=osszeg+B;


        end

        fn=abs(c-osszeg);


end

    function fn=s4()

    %% random pontok Simpsonhoz

    hossz=1;

    eleje=0;

    a(1)=0;

    i=2;

    while 1

        k=rand/50;

        if (hossz-2*k)<0
```

```
            a(i)=eleje+(hossz/2);
            a(i+1)=1;
            break;
        else
            eleje=eleje+k;
            a(i)=eleje;
            i=i+1;
            eleje=eleje+k;
            a(i)=eleje;
            i=i+1;
            hossz=hossz-2*k;
        end
    end
    fn=a;
end
    function fn=s13(x,pont)
    %% Simpson random pontokra
    % x=s2();
    y=poly2sym(x);
```

```
    f=inline(y);

    % pont=s4();

    e=s3(x);

    z=length(pont);

    k=0;

    i=1;

    while i<z

        h=pont(i+1)-pont(i);
        a=pont(i);
        b=pont(i+1);
        c=pont(i+2);

        k=k+h*((f(a)+4*f(b)+f(c))/3);

        i=i+2;


    end

    fn=abs(k-e);

end

    function fn=s12()

x=s2();

d=s3(x);

y=poly2sym(x);
```

```
f=inline(y);

i=2;

while i<1025

    k=0;
    h=1/i;

    for j=1:i/2

        a=(2*j-2)*h;
        b=(2*j-1)*h;
        c=2*j*h;

        k=k+h*((f(a)+4*f(b)+f(c))/3);

    end

    disp(abs(d-k))
    i=i*2;

end

disp(y);

end

    function fn=l11(x,pont)

    % Lagrange-interpoláció
    % "véletlenszerű" pontok
    % "véletlenszerű" függvény

    % x=s2();

    y=poly2sym(x);
```

```
f=inline(y);

c=s3(x);

% pont=s1();


for i=1:length(pont)

    s(i)=f(pont(i));

end

L(1)=0;
z=2;

for i=2:length(pont)

    if pont(i)~=pont(i-1)

        L(z)=0;
        z=z+1;

    end

end

for i=1:length(pont)

    I=[1];
    K=1;

    for j=1:length(pont)

        if pont(i)~=pont(j)

            T(1)=1;
            T(2)=pont(j)*(-1);
```

```
                K=K*(pont(i)-pont(j));
                I=conv(I,T);

            end

        end

        H=s(i)/K;
        M=I*H;

        for g=1:length(L)

            L(g)=L(g)+M(g);

        end

    end

    O=s3(L);

    fn=abs(O-c);

    disp(y);

    disp(length(pont));


end

    function fn=t13()

x=s2();

c=s3(x);

y=poly2sym(x);
```

```
f=inline(y);

i=1;

while i<1025

    k=0;
    h=1/i;

    for j=1:i

        a=(j-1)*h;
        b=j*h;

        k=k+h*((f(a)+f(b))/2);

    end

    disp(abs(c-k))
    i=i*2;

end

disp(y);

end

    function fn=t12(x,pont)

% x=s2();

y=poly2sym(x);

f=inline(y);

% pont=s1();

N=length(pont);
```

```
h=1/(N-1);

z=0;

e=s3(x);

for i=1:N-1

        a=pont(i);
        b=pont(i+1);

        z=z+h*((f(a)+f(b))/2);

end

fn=abs(z-e);

end

    function fn=s2()

    while 1

        darab=rand*10;

        darabegesz=round(darab);

        if(darabegesz>0)
            break
        end
    end

    for i=1:darabegesz

        X=rand*10;
        P(i)=round(X);
```

```
    end

    fn=P;

end

    function fn=s2()

    while 1

        darab=rand*10;

        darabegesz=round(darab);

        if(darabegesz>0)
            break
        end
    end

    for i=1:darabegesz

        X=rand*10;
        P(i)=round(X);

    end

    fn=P;

end
```