Budapesti Műszaki és
Gazdaságtudományi Egyetem

Közlekedésmérnöki és Járműmérnöki
Kar

Dr. Bicsák György

MÉRNÖKI SZÁMÍTÁSOK

XY Kiadó

BME Közlekedésmérnöki és Járműmérnöki Kar

32708-2/2017/INTFIN számú EMMI által támogatott tananyag

© Dr. Bicsák György, 2019

ISBN XXX-XXX-XXX-XXX-X

Kiadja az XY Kiadó Kft.

Készült az XY Bt. nyomdájában

Tartalomjegyzék

1 Bevezető

Habár az egyetemi tanulmányaink alatt viszonylag széleskörű matematikai ismeretek kerülnek átadásra, azok gyakorlati használhatósága sokszor nem teljesen egyértelmű, vagy csak későbbi tanulmányok során kerülnek elő. Ráadásul a mérnöki gyakorlatban a különböző, analitikus módon történő feladatmegoldások igen körülményesek, bonyodalmasak lehetnek, minden erőfeszítést egy olyan pontosságú megoldásért téve, melyre igazából nem is biztos, hogy van szükség. A matematikai „korrekt megoldás" helyett elég sűrűn mondjuk azt a való életben, hogy a kapott eredmény nem 100%-ban pontos, de a célnak megfelelő. Hasonló elv mentén kerültek megalkotásra a különböző numerikus módszerek, melyeknek kíván megágyazni ez a tárgy/jegyzet. Azokat az eseteket szintén ide kell vennünk, melyeknek matematikailag nem meghatározható a megoldása, viszont valamit mégis ki kell találni (például végeselemes, véges térfogatos szimulációk esetén).

Az egyes numerikus módszerek lényege, hogy az egzakt, olykor igen bonyolult és műveletigényes matematikai megoldás helyett egy aritmetikai egyenletekből felépülő algoritmus segítségével a pontos eredményt csak megközelítsük valamilyen pontossággal. A numerikus módszerek alapvetően – miután már megértettük őket – igen egyszerűnek tűnhetnek, viszont az elengedhetetlen, hogy megértsük az egyes módszerek matematikáját is: vagyis hogy miért tehetjük meg azt, hogy numerikus módszerekkel oldjuk meg a feladatainkat.

Természetesen a numerikus módszerek esetében sem feltétlenül fenékig tejfel az élet, hiszen minél pontosabban szeretnénk közelíteni a megoldásunkat, annál nagyobb műveletszámra van szükségünk, és ez például szimulációk esetében akár igen hosszadalmas folyamat lehet. Zárójelben meg kell jegyezni, hogy ezekkel a szimulációkkal még mindig jobban járunk a legtöbb esetben, mintha a szimulált jelenségeket valós méréseknek vetnénk alá. Ahhoz, hogy a későbbi tanulmányaink során megjelenő szimulációs eszközök (FEM, VEM, CFD …) érthetőbbek legyenek, a félév során kezdésként megnézzük azt, hogy numerikus algoritmusok használata során milyen hibákkal kell számolnunk, mi okozza őket, milyen hatása lehet a végeredményre, hogyan terjedhetnek tovább. A további fejezetek 3 fő témakörbe sorolhatók:

- Görbeillesztési eljárások
- Numerikus deriválás és integrálás
- Optimalizációs eljárások.

Maga a numerikus módszereknek a szakirodalma ma már igen kiterjedt, főleg az angolszász jegyzetek és könyvek között nagyon sok gyakorlat-orientált kiadványt lehet találni. A legtöbb jegyzet jellemzője, hogy az elméleti áttekintés során nem igen taglalják az egyes algoritmusok elméletét. Tekintve, hogy egy B.Sc.-s 3. féléves tantárgyról beszélünk, így a tantárgy tematikájának és témaköreinek meghatározása során azok kerültek összegyűjtésre, melyek a mérnöki gyakorlatokban a leggyakrabban fordulnak elő. Természetesen előfordulhat, hogy valaki sosem találkozik majd ilyen problémákkal, és mérnöki élete során nem fog kitérni a 4 alapművelet bűvköréből, viszont az is lehetséges, hogy ennél sokkal komplexebb problémákkal kell megküzdenie nap mint nap.

Szintén a gyakorlati alkalmazhatóság miatt esett a választás az Excel programra, mellyel az egyes példák demonstrálása történik. Ha MATLAB-ot nem is fog mindenki használni élete során, Excelt szinte biztosan. Ráadásul az Excelből sokkal több is kihozható egyszerű táblázatoknál és alapfüggvények használatánál (lásd: http://www.excelunusual.com/), főleg ha a VBA használata is ismert. Ez utóbbi alapjainak elsajátítására nem feltétlenül van lehetőség órai keretek között, ezért egyrészt az első féléves Számítástechnika tárgyra támaszkodunk, illetve Pusztai Pál: Excel Programozás (SZIE - 2014) elektronikus jegyzetének autodidakta átolvasása erősen ajánlott. A félév után remélhetőleg a tárgyat teljesítők az Excel sokkal szélesebb körben való alkalmazására lesznek képesek, akár saját függvények írásával.

Természetesen a félévhez jó szórakozást kívánok!

2 Hibák és közelítések

A numerikus módszerek olyan eljárások, melyek egy matematikai probléma megoldását nem az egzakt módon, hanem közelítéssel, véges számú lépések végrehajtásával adja meg egy előre meghatározott pontossággal. Habár egyszerűbb számítások akár számológépen is kivitelezhetők, a legtöbb numerikus módszer megoldásához számítógépre van szükség. Mindezek ellenére sok esetben még így is gyorsabb megoldást szolgáltatnak az egzakt módnál, tekintve hogy egyszerű aritmetikai műveletekből tevődnek össze, és sok esetben limitált végrehajtással.

A numerikus módszerek alatt általában egy erős matematikai háttérrel rendelkező algoritmust értünk, mely tulajdonképpen nem más, mint egy olyan egyértelmű utasításrendszer, mely a meghatározza a felhasznált műveletek tartalmát és sorrendjét, és megfelelő kiinduló adatokból a kívánt eredményt szolgáltatja. Ekkor jön a képbe a „megfelelőség" fogalma. Mivel egy-egy numerikus módszerrel csak közelíteni tudjuk az egzakt megoldást, ezért szemben a matematikai végeredménnyel, itt meg kell elégednünk annak csak valamekkora pontossággal való közelítésével. Ráadásul, ha eleve csak közelítünk, akkor különösen fontos lesz a bemenő adatok pontossága, hiszen egy valamekkora szórással rendelkező adathalmaz nagymértékben tudja befolyásolni az algoritmus pontosságát. Erről a következő fejezetben lesz szó részletesebben. A másik probléma, hogy mivel csak közelítjük a pontos megoldást, tisztában kell lennünk azzal, hogy annak mekkora a hibája, és az milyen módon tudja befolyásolni a végeredményünket. Ehhez először tekintsük át a különböző hibatípusokat, azok forrásait és tovaterjedésüket!

2.1 Hibák és forrásaik:

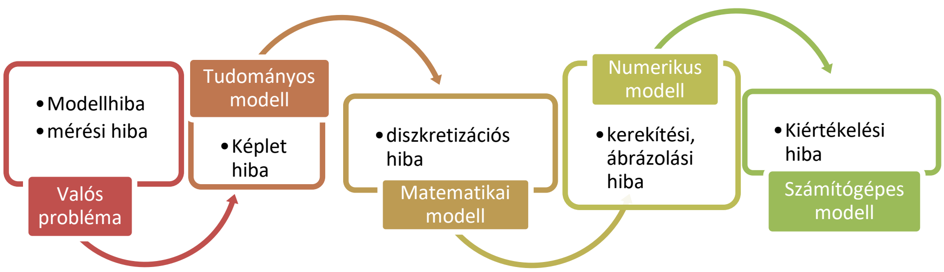
A modellalkotás során számos lépés vonhat magával hibagenerálást. A modell a valóság valamilyen módon történő leképzése, leegyszerűsítése, így ez mindenképpen bele fog kerülni a számításainkba. Sok esetben pedig pont a hiba az, ami lehetővé teszi egy matematikai modell egyenletének megoldását. Egy modellalkotási folyamat során a következő hibák merülhetnek fel: [1]

1. Modellhiba

- A valóság és a közelítő modell leegyszerűsítéséből származó hibát nevezzük modellhibának. Például egy statikai modell esetén azt feltételezzük, hogy a terhelt rúd végtelenül merev, és nem mozdul el, pedig tudjuk, hogy nagyon is deformálódik, ebből adódóan el is tud mozdulni.

2. Mérési hiba
- Ezt az Általános Járműgéptan óta talán nem kell részletezni, hogy mivel egy mérésben számtalan zavaró tényező jelenik meg, így abba mindig csúszhat hiba. Ezért tudjuk nagyon jól, hogy „egy mérés nem mérés”.
3. Képlet hiba
- Egy képlet kezelhetőségének érdekében történő egyszerűsítése során (például egy Fourier sornak nem végtelen sok tagját használjuk, hanem csak az első 3-at), ekkor képlet hibáról beszélünk.
4. Diszkretizációs hiba
- A numerikus módszerek/eljárások sajátja, mivel ilyenkor folytonos függvényeket rácsfüggvénnyel, deriváltat differenciahányadossal, integráltat pedig szorzatösszeggel közelítjük.
5. Kerekítési és ábrázolási hiba
- A számítógépbe bevitt adatok ábrázolása során a bináris logika miatt az adatok torzulhatnak, ekkor keletkezik az ábrázolási hiba. A számítások során alkalmazott kerekítésekből származó hibák pedig a kerekítési hibát adják (például a π szám ábrázolásához a világ összes háttértára sem lenne elég, ezért kerekíteni kell annak értékét).
6. Kiértékelési hiba
- Az elkészített, lefuttatott modell időnként konvergált, pontosnak tűnő eredményt ad, de a peremfeltételek megadása, a modell összetevőinek definiálása, különböző hatások figyelembe vétele során elkövetett hibák miatt az eredmény mégsem pontos, amelyet a kiértékelés során vagy észlelünk, vagy nem. Az észlelt hiba természetesen lehet elfogadható vagy nem elfogadható, jelenléte miatt a valóságtól mégis eltérő eredmény születik.

Az egyes modellalkotási lépéseket és a közöttük előforduló hibaforrásokat a 1. ábra szemlélteti.



1. ábra: A modellalkotás és az egyes lépések közötti hibaforrások [1]

2.1.1 Bináris és hexadecimális számok

Általános célokra a tízes számrendszer használata terjedt el. Ezt talán nem is kell túlmagyarázni, de biztonság kedvéért az ebben a számrendszerben kifejezett számok a következőképpen épülnek fel:

$$642 = [6 * 10^2 + 4 * 10^1 + 2 * 10^0]_{10}$$

Persze az egyszerűség érdekében (főleg nagyon nagy és nagyon kicsi számok esetén) alkalmazzuk a tízes számrendszerben felírt számok normál alakját a következő formában:

$$\pm d_1, d_2 d_3 \ldots d_m * 10^k, \ \ 1 \leq d_1 \leq 9, \ \ 0 \leq d_2, d_3, \ldots, d_m \leq 9$$

Például: $642 = 6{,}42 * 10^2$

ahol az első tag a karakterisztika, a második a mantissza (más néven exponens). Ezt viszont már első félév óta tudjuk, hogy ezt az alakot nem csak normál alaknak hívjuk, hanem lebegőpontos ábrázolásként is hivatkozhatunk rá. Illetve fontos megemlíteni, hogy a magyar és angol (külföldi) ábrázolása a normál alaknak és általában véve a számoknak tartalmaz egy apró, mégis igen bosszantó különbséget: míg itthon a tizedesvessző az ténylegesen egy vessző, addig szinte mindenhol máshol pontként kezelik, amiből adódóan minden programozási nyelvben, és a legtöbb szoftverben, ha egy tizedes törtet vesszővel írunk be, azt a program nem tudja értelmezni. Csak a móka kedvéért persze vannak olyan szoftverek is, melyek egyes modulokon belül egyszer az egyik, másszor a másik beviteli módot alkalmazzák, csak hogy érezzük a törődést.

A számítógépeken belül viszont az igaz/hamis logikából következően nem a tízes számrendszer, hanem a kettes (bináris) számrendszer kerül felhasználásra. Ahogy már ezt kiválóan tudjuk ekkor a jól bevált számunk a következőképpen írható fel:

$$642 = [1 * 2^9 + 0 * 2^8 + 1 * 2^7 + 0 * 2^6 + 0 * 2^5 + 0 * 2^4 + 0 * 2^3 + 0 * 2^2 + 1 * 2^1 + 0 * 2^0]_2$$

vagyis a bináris alak: $642 = [1010000010]_2$

A szintén elterjedten alkalmazott a számítástechnikában a 16-os számrendszer. Most már konzekvensen:

$$642 = [2 * 16^2 + 8 * 16^1 + 2 * 16^0]_{16} = [282]_{16}$$

Mivel itt egy karakter 16 számot jelenthet, így a 0..9 számjegyek mellett megjelennek a betűk is A-tól F-ig, vagyis például:

$$30 = [1 * 16^1 + 14 * 16^0]_{16} = [1E]_{16}$$

A 16-os számrendszer elsőre kicsit felvágósnak tűnhet, alkalmazásának oka, hogy ez tulajdonképpen a kettes számrendszer kiterjesztése, hiszen $2^4 = 16$, vagyis minden kettes számrendszerben felírt számnégyesre jut egy digit 16-os számrendszerben. Például $D = [1101]_2$ vagy $9 = [0101]_2$.

2.1.2 Lebegőpontos ábrázolás és lekerekítési hibák

Mivel a számítógépek csak véges számú digitet képesek tárolni és megjeleníteni, így a számokat oly módon érdemes tárolni, hogy az csak egy adott (fix) mennyiségű digitet használjon erre a célra. Ennek következtében a számítástechnikában alapvetően két fajta ábrázolás terjedt el: a fixpontos és a lebegőpontos számábrázolás. Az első esetben a számok tárolásakor meghatározásra kerül, hogy hány tizedesjegyet szeretnénk tárolni, például: 4,0000; -129,1012; 0,0023. Ekkor a 4 tizedesjegy akkor is felhasználásra kerül, ha egész számot akarunk eltárolni. A lebegőpontos ábrázolás esetén pedig meghatározott számú digiteket tárol a karakterisztikából, így például:

$$4 * 10^0; \quad -1{,}2910 * 10^2; \quad 2{,}3000 * 10^{-3}$$

Általános alakban a lebegőpontos számábrázolás bármilyen számot a következő alakban ad meg:

$$(-1)^S \, M \, p^k \tag{2.1.}$$

ahol S az előjel, M jelöli a mantisszát, p pedig az alapot, továbbá k a karakterisztikát. Továbbra is igaz, hogy $1/p \leq M \leq 1$.
Eltérő architektúrák esetén más és más lebegőpontos ábrázolási módot alkalmazhatunk. Jelenleg az Institute of Electrical and Electronics Engineers (IEEE) által definiált IEEE 754 szabvány a mérvadó, mely már egy 30 éves szabvány. Ezzel egy 32 bites (egyszeres pontosságú) számot a következőképpen alkalmazhatunk: [2]

$$szám = (-1)^S \, (1.M)\left(2^{k-127}\right) \tag{2.2.}$$

Az előjel 1 bit hosszúságú, értéke pedig 0 pozitív számok esetén, 1 negatív számok esetén. Ezután következik a 8 bit hosszúságú karakterisztika, mely kijelöli a kettedes pont helyét a számban. Legtöbbször nullpontú vagy többletes formában tárolják. Amennyiben a karakterisztika mező *k* bit hosszúságú, akkor az eltolási érték $e = 2^{k-1} - 1$. Egyszeres pontosság esetén ez: $e = 127$.

A következő a mantissza, ami 23 bit hosszúságú és egy egészre normált törtszám. Első számjegye mindig 1, amely bitet nem tárolja a formátum, csak a törtrészt. Az eltárolt szám a következőképpen számítható ki:

$$szám = (-1)^S \, (1 + M)\left(2^{k-127}\right) \tag{2.3.}$$

ahol *S* az előjelbitet jelöli, *M* a mantisszát, *k* a karakterisztikát és végül *e* az eltolást.

A mantissza számára fenntartott bitek száma a számábrázolás pontosságát határozza meg, míg a karakterisztika mérete az ábrázolható számok nagyságrendjét. [2]

Példa:

Melyik számot jelenítettük meg 32 biten a következő alakban:

1 10000110 10100000000000000000000 ?

Megoldás:

Az előbbi gondolatmenet alapján (mondhatnánk triviálisan):

$$s = 1$$
$$k = 10000110_2 = 134_{10}$$
$$M = 0.101_2 = 0{,}625_{10}$$

Így:

$$x = -1{,}625 * 2^7 = -1{,}625 * 128 = -208$$

Vagyis a kezdeti csúf-gonosz szám mögött a -208 bújik meg.

2.1.3 Abszolút és relatív hibák

Mérnöki gyakorlatban mind a mérés, mind a modellalkotás, szimuláció-kiértékelés során a keletkezett eltérést, hibát valamilyen módon számszerűsíteni kell, függetlenül a hiba keletkezésének forrásától. A hibaszámítás elvégzésére alapvetően két hibatípust alkalmazunk: az abszolút és relatív hibát. Ha a mért/számított adatot $\tilde{x}$ jelöli, a pontos adatot pedig x, akkor a két hiba a következőképpen adható meg:

Abszolút hiba:

$$\delta_a = x - \tilde{x} \qquad (2.4.)$$

Relatív hiba:

$$\delta_r = \frac{abszolút\ hiba}{valós\ érték} = \frac{\delta_a}{x} = \frac{x - \tilde{x}}{x},\quad x \neq 0 \qquad (2.5.)$$

Amennyiben a valós érték mégis 0 lenne, akkor a relatív hibát nem értelmezzük. Általában véve a relatív hibának nagyobb gyakorlati jelentősége van, mint az abszolút hibának, ezért legtöbb esetben a relatív hiba számítása történik meg.

2.1. Példa

Vegyünk két esetet: először elektromos áramerősséget mérünk, amire $\tilde{x}_1 = 0{,}004\ A$, míg a valós érték $x_1 = 0{,}005\ A$. A második esetben feszültséget mérünk, ahol a mért érték $\tilde{x}_2 = 1315\ V$, míg a tényleges érték $x_2 = 1331\ V$. Mekkora az abszolút és relatív hiba a két esetben?

Megoldás:

Abszolút hibák:

$$\delta_{a1} = x_1 - \tilde{x}_1 = 0{,}005\ A - 0{,}004\ A = 0{,}001\ A$$
$$\delta_{a2} = x_2 - \tilde{x}_2 = 1331\ V - 1315\ V = 16\ V$$

Ugye, hogy mennyivel kisebbnek tűnik a 0,001 A, mint a 16 V a mért különbségben? De nézzük csak meg, hogy mit jelent ez a relatív hibák tükrében!

$$\delta_{r1} = \frac{x_1 - \tilde{x}_1}{x_1} = \frac{0{,}005\ A - 0{,}004\ A}{0{,}005\ A} = 0{,}2$$
$$\delta_{r2} = \frac{x_2 - \tilde{x}_2}{x_2} = \frac{1331\ V - \ 1315\ V}{1331\ V} = 0{,}012$$

Tehát míg az áramerősség esetében a 0,001 A abszolút hiba az 20%-os relatív hibának felel meg, addig a feszültség mérése esetében a 16 V abszolút hiba az csak 1,2%-os relatív hibát jelent. Ezért nem mindegy, hogy ha az ember eggyel, vagy 50%-kal kevesebb pofont kér, akkor mennyi az a szám, amiből kiindulunk.

Hibahatár

Ha egy-egy eltérésre vagyunk kíváncsiak, akkor az abszolút hiba ad erre valamekkora megközelítést, de ha már több eset átlagára vagyunk kíváncsiak, célszerű az abszolút hiba abszolút értékét venni, így kiküszöbölve, hogy a pozitív és negatív tagok kiüssék egymást. A következő felmerülő kérdés, hogy mi az az abszolút hiba, ami még belefér az adott számításba, és melyik az, amelyik nem? Míg az abszolút hiba alsó értéke egyértelműen zéró lehet, addig a felső határ akár végtelent is felvehet, így érdemes egy felső korlátot bevezetni, az úgynevezett hibahatárt:

$$|\delta_a| = |x - \tilde{x}| \leq \alpha \tag{2.6.}$$

Meg kell jegyezni, hogy α akármennyire is jóképű, nem fog becslést adni az abszolút hiba értékére, csak egy limitet ad. Hasonlóképpen tudunk egy hibahatárt definiálni a relatív hibák értékére:

$$|\delta_r| = \frac{|x - \tilde{x}|}{|x|} \leq \beta \tag{2.7.}$$

2.1.4 Hibaterjedés

Az egyes hibák típusainak megismerése után a következő kérdés az, hogy ezek a hibák hogyan „fejlődnek” tovább, ezzel hogyan befolyásolják egy számítás/szimuláció végeredményét. Tekintve, hogy a numerikus számítások során nagyrészt összeadási/kivonási és szorzási/osztási műveletek kerülnek végrehajtásra, a vizsgált hatások egyértelműen meghatározhatók lesznek.
A következőkben tegyük fel, hogy $\tilde{x}_1$ és $\tilde{x}_2$ a valós x_1 és x_2 értékek közelítései. Ezen párosok abszolút és relatív hibái:

$$|\delta_{a1}| \leq \alpha_1; \; |\delta_{a2}| \leq \alpha_2; \; |\delta_{r1}| \leq \beta_1; \; |\delta_{r2}| \leq \beta_2 \tag{2.8.}$$

A felső hibahatár az abszolút hibára vonatkoztatva összeadás és kivonás esetén az egyes abszolút hibahatárok összege/különbsége lesz:

$$|\delta_a| = |(x_1 \pm x_2) - (\tilde{x}_1 \pm \tilde{x}_2)| \leq \alpha_1 + \alpha_2 \tag{2.9.}$$

A relatív hibák esetében másképpen fog terjedni a hiba:

$$|\delta_r| = \left|\frac{x_1 x_2 - \tilde{x}_1 \tilde{x}_2}{x_1 x_2}\right| \leq \beta_1 + \beta_2 \tag{2.10.}$$

Ez könnyen belátható, hiszen ha mondjuk leegyszerűsítve a problémát két mennyiség összeszorzásának relatív hibáját szeretnénk számolni, akkor a következőképpen járnánk el:

$$|\delta_r| = \left|\frac{x_1 x_2 - \tilde{x}_1 \tilde{x}_2}{x_1 x_2}\right| = \left|\frac{x_1 x_2 - [x_1 - (\delta_{a1})][x_2 - (\delta_{a2})]}{x_1 x_2}\right| = \tag{2.11.}$$

$$= \left|\frac{-(\delta_{a1})(\delta_{a1}) + (\delta_{a1})x_2 + (\delta_{a2})x_1}{x_1 x_2}\right|$$

Ez utolsó kifejezésben pedig nyugodtan élhetünk azzal a közelítéssel, hogy $-(\delta_{a1})(\delta_{a1}) \approx 0$, hiszen összehasonlítva a másik két taggal, ennek nagyságrendje sokkal kisebb. Így pedig azt kapjuk, hogy:

$$|\delta_r| = \left|\frac{(\delta_{a1})x_2 + (\delta_{a2})x_1}{x_1 x_2}\right| = \left|\frac{(\delta_{a1})}{x_1} + \frac{(\delta_{a2})}{x_2}\right| \leq \left|\frac{(\delta_{a1})}{x_1}\right| + \left|\frac{(\delta_{a2})}{x_2}\right| \tag{2.12.}$$

$$\leq \beta_1 + \beta_2$$

Ezzel tulajdonképpen hirtelen felindulásból be is bizonyítottuk igazunkat.
Hasonlóképpen – de most nagy megkönnyebbülésre bizonyítás nélkül – osztásra a relatív hibaterjedés a következőképpen alakul:

$$|\delta_r| = \left|\frac{x_1/x_2 - \tilde{x}_1/\tilde{x}_2}{x_1/x_2}\right| \leq \beta_1 + \beta_2 \tag{2.13.}$$

2.1.5 Két közel nulla szám kivonása

A műveletek elvégzése során két esetben egy viszonylag egyszerű művelet is nagy pontatlansághoz vezethet:

- kis nagyságrendű számmal történő osztás
- két majdnem egyenlő számmal történő kivonás.

Természetesen, ha ez utóbbi kivonás egy tört nevezőjében történik, akkor könnyen az első esethez vezethet. Vegyünk például két számot, melyeket jelöljenek S_1 és S_2, továbbá lebegőponti ábrázolásukban az első k digit megegyezik:

$$FL(S_1) = 0, d_1 d_2 \dots d_k a_{k+1} \dots a_m \times 10^p$$
$$FL(S_2) = 0, d_1 d_2 \dots d_k b_{k+1} \dots b_m \times 10^p$$

Minél magasabb k értéke, természetesen S_1 és S_2 annál inkább egyenlők lesznek. A kivonás után a következő alakot kapjuk:

$$FL(FL(S_1) - FL(S_2)) = 0, c_{k+1} \dots c_m \times 10^{p-k}$$

ahol $c_{k+1} \dots c_m$ az $(a_{k+1} - b_{k+1}) \dots (a_m - b_m)$ kivonások eredményei. Igaz, elsőre ijesztőnek tűnhet a kifejezés, de ha közelebbről megnézzük, rájöhetünk, hogy kifejezetten barátságos, hiszen nem történt más, minthogy a mantissza $m - k$ elemre csökkent az eredeti S_1 és S_2 számokhoz képest, ahol m részből állt a mantissza, k elem pedig kvázi kiesett. Ezzel viszont nagymértékben felerősítjük a kerekítési hibát, mely innentől fogva befolyásolni fogja a számításunkat. Ez a hatás eliminálható, ha a problémát átgondoljuk. Nézzünk erre egy példát!

2.2. Példa

Legyen adott a következő másodfokú egyenlet: $x^2 + 50x + 4 = 0$, melynek gyökei: $x_1 = -0{,}080128411$ és $x_2 = -49{,}91987159$.

Megoldás:

Már jól tudjuk, hogy a másodfokú egyenletek megoldására van egy jól bevált képletünk:

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Ebben az esetben $b^2 \gg 4ac$, vagyis a gyök alatti részről elmondható, hogy: $\sqrt{b^2 - 4ac} \cong b$. Vagyis x_1 számításakor majdnem egyenlő számokat vonunk ki egymásból, ami például egy 4 digites lekerekítés esetén a következő hibát adja:

$$FL(x_1) = \frac{-50,00 + \sqrt{50,00^2 - 4(1,000)(4,000)}}{2(1,000)} = \frac{-50,00 + 49,84}{2,000} = -0,060$$

Illetve hasonlóképpen a másik gyök:

$$FL(x_2) = \frac{-50,00 - \sqrt{50,00^2 - 4(1,000)(4,000)}}{2(1,000)} = \frac{-50,00 - 49,88}{2,000} = -49,94$$

Ezekre az értékekre a relatív hibák nagysága:

$$|\delta_{r1}| = \frac{|x_1 - FL(x_1)|}{|x_1|} \cong 0,2503 = 25,03\%$$

$$|\delta_{r2}| = \frac{|x_2 - FL(x_2)|}{|x_2|} \cong 0,0004 = 0,4\%$$

Vagyis x_1 jóval nagyobb relatív hiba értéket fog produkálni, mint x_2, hiszen ott a két közel egyforma számot összeadjuk és nem kivonjuk egymásból, így ott ez nem okoz problémát. Az igazi baj viszont, hogy ugyanazon képlet ugyanazon problémára tud adni 0,4% vagy 25% relatív hibát, amire még egy ilyen egyszerű esetben is tekintettel kell lennünk.

Hogy úrrá legyünk a káoszon, közelítsünk másképp a feladathoz. A megoldóképlet mellett azt is tudjuk (elvileg), hogy az $ax^2 + bx + c = 0$ egyenletet kielégíti a $x_1 x_2 = c/a$. Ha $FL(x_2)$ értékéből kiindulva számítjuk $FL(x_1)$-et, akkor:

$$FL(x_1) = \frac{c}{aFL(x_2)} = \frac{4,000}{(1,000)(-49,94)} = -0,080$$

Ekkor a relatív hiba:

$$|\delta_{r1}| = \frac{|x_1 - FL(x_1)|}{|x_1|} \cong 0,0016 = 0,16\%$$

Vagyis egy drasztikus javulást (24,87%) értünk egy végtelenül egyszerű javítással az eredeti számítási módszerhez képest.

2.2 Iteratív módszerek

A numerikus módszerek tulajdonképpen olyan előre meghatározott algebrai és logikai matematikai műveletek meghatározott sorrendű elvégzését jelenti, amivel egy adott probléma analitikus módon történő egzakt megoldása helyett annak közelítő megoldását adja ki. Ezen utasításrendszer, melyek adott műveletek tartalmát és sorrendjét egyértelműen meghatározzák, és adott kiinduló adatokból a keresett eredményt adják ki, algoritmusoknak nevezzük.

Az algoritmusok leírásához pedig a leghatékonyabb leírási módszer a program kóddal való bemutatás. A kód tartalmazza a bemeneteket (input), a szükséges műveleteket és a kimeneteket (output). Általában véve kétfajta írásjelet használnak az algoritmusok leírásában: az aktuális lépés befejezését a pont (.) jelöli, míg ha az adott lépés még mindig folyamatban van, azt pontosvesszővel (;) jelölik. Egy algoritmust stabilnak nevezünk, ha a bemenő adatokban bekövetkező kis változás a végeredményben (output) is kis változást idéz elő. Ellenkező esetben az algoritmus instabil.

Egy iteratív módszer alatt az olyan folyamatokat értjük, melyek találgatással kezdődnek, majd a probléma megoldását szukcesszív approximációval számolja egészen addig, míg a közelítés kielégítő nem lesz. Az ilyen iteratív módszerek felhasználása igen sokrétű lehet, algebrai egyenletek megoldásától kezdve az egyenletrendszerek megoldásán át egészen a differenciál egyenletek megoldásán kívül számos alkalmazása létezik. Ezen jegyzet (és az erre a tárgyra épülő Járműmérnöki Matematika tantárgy) ilyen módszerek bemutatását, és azok alkalmazhatóságát taglalja.

Az iteratív módszerek egyik legfontosabb kérdése a leállítási feltétel. Két módszert használunk általában:

1. amikor egy megszakítási feltétel igazzá válik
2. amikor egy maximum iterációs lépésszámot elérünk.

Elvileg a leállítási feltételnek ellenőriznie kell, hogy számított approximáció az előre definiált tolerancián belül van-e. Gyakorlatban viszont a pontos érték nem mindig áll rendelkezésre, így a leállítási feltétel sokszor az egyes iteratív lépések által számolt értékek közötti változás nagyságának egy bizonyos tűréshatáron belül-lévőségét tudja csak ellenőrizni. Ideális esetben az algoritmus viszonylag gyorsan megfelel az első típusú feltételnek. Ha ez nem így lenne, akkor pedig muszáj beállítani egy maximális lépésszámot, enélkül örökké futna a kódunk.

2.3. Példa: [3]

Nézzünk egy matematikai példát! Közelítsük e^{-2} értékét hét tizedesjegyig, $\epsilon = 10^{-6}$ toleranciával!

Megoldás:

Határozzuk meg az első $n+1$ tagot Maclaurin sor használatával, melyet az $f(x) = e^x$ függvényre írunk fel egy *n*-ed fokú Taylor polinommal:

$$T_n(x) = \sum_{i=0}^{n} \frac{1}{i!} x^i$$

Ha az e^{-2}-t szeretnénk közelíteni ezen egyenlettel a legkisebb lehetséges n felhasználásával, akkor:

$$|e^{-2} - T_n(-2)| < \varepsilon$$

Ez lesz egyben a leállítási feltétel is.
A megoldó algoritmus ekkor a következőképpen építhető fel:

Bemenet:		$x = 2, \varepsilon = 10^{-6}, N = 20$	
Kimenet:		e^{-2} közelítése ε pontossággal, vagy „hiba" üzenet	
1.	lépés	Állítsunk be kezdetben: $n = 0$	
2.	lépés	$Tval = e^{-x}$ $Term = 1$ $Psum = 0$ $Sgn = 1$	 Valós érték Kezdeti összeg Váltakozó előjel kezdeti beállítása
3.	lépés	$While\ n \leq N, do\ Lépés\ 3 - Lépés\ 5$	
4.	lépés	$Psum = Psum + Sgn * Term/n!$ $Stop$	Leállítási feltétel
5.	lépés	$n = n + 1$ $Sgn = -Sgn$ $Term = Term * x$	n frissítése Előjel *Term* frissítése
6.	lépés	$Output(failure)$ $Stop$	
Vége			

A hét tizedesjegy pontossággal megadva: $e^{-2} = 0{,}1353353$. Ha a maximum iterációk számát 20-nak, így valószínűleg hibaüzenettel fog megállni a program, mivel ennyi lépés alatt nem biztos, hogy sikerül a hibahatáron belül kerülni. Ha a közelítés és pontos érték közötti különbség mégis kisebb lenne, mint az előírt tolerancia, a leállítási feltétel szerint megállításra kerülne a program. Ekkor a kimenetek: n és e^{-2} közelítése. A programot, ha kipróbáljuk MATLAB-ban, vagy Excel VBA-ban (később erre is sor kerül), akkor kiderül, hogy 14 iterációs lépés szükséges a leállítási feltétel eléréséhez, vagyis $n = 13$. Ekkor a közelítő érték: $e^{-2} \cong 0{,}1353351$, ami $\delta_a = 0{,}2 * 10^{-6}$ abszolút hibát jelent a pontos értékhez képest.

2.2.1 Alap iteratív módszer

Az alap iteratív módszerek ismétlődő behelyettesítéseket használnak. Például ha van egy $f(x)$ függvényünk, és x_0 kezdeti értékünk, akkor $x_1, x_2, \dots x_{n+1}$ értékeit a következő módon tudjuk generálni:

$$x_{n+1} = f(x_n) \tag{2.14.}$$

$n = 0,1,2, \dots$, és x_0 értékek ismertek.

A lefutás szempontjából nagyjából 3 lehetőség van:

- az iteráció gyors ütemben konvergál
- az iteráció lassú ütemben konvergál
- az iteráció nem konvergál.

Az, hogy miképp fut le a program, nagymértékben függ olyan tényezőktől, mint a $f(x)$ függvény jellege vagy x_0 kezdőérték.

2.2.2 Konvergencia ráta

Vegyünk egy $\{x_n\}$ sorozatot, mely x-hez konvergál. Ekkor az *n*-edik iterációs lépésnél az abszolút hiba értéke:

$$\delta_{a,n} = x - x_n, \quad n = 0,1,2, \dots$$

Ha létezik egy olyan *R* szám és egy *K* konstans, amire:

$$\lim_{n \to \infty} \frac{|\delta_{n+1}|}{|\delta_n|^R} = K \tag{2.15.}$$

akkor azt mondjuk, hogy az iterációs algoritmus konvergencia rátája *R*. Két alapvető konvergencia típust különböztetünk meg egymástól: a lineárist és négyzetest. A konvergencia lineáris, ha $R = 1$, vagyis:

$$\lim_{n \to \infty} \frac{|\delta_{n+1}|}{|\delta_n|} = K \neq 0 \tag{2.16.}$$

A konvergencia négyzetes, ha $R = 2$, vagyis:

$$\lim_{n \to \infty} \frac{|\delta_{n+1}|}{|\delta_n|^2} = K \neq 0 \tag{2.17.}$$

A konvergencia ráta nem mindig integer szám. Ha vesszük például a felező módszert, annak a konvergencia rátája $\frac{1}{2}\left(1 + \sqrt{5}\right) \cong 1{,}618$. [3]

2.4. Példa: [3]

Határozzuk meg a következő probléma konvergencia rátáját:

$$x_{n+1} = \left(\frac{1}{2}\right) x_n, \quad n = 0,1,2,\dots, \qquad x_0 = 1$$

Megoldás:

Mivel ha $n \to \infty$, akkor $x_n = \left(\frac{1}{2}\right)^n \to 0$, természetesen $x = 0$ határral. Ekkor az *n*-edik lépésnél az abszolút hiba:

$$\delta_{a,n} = x - x_n = 0 - \left(\frac{1}{2}\right)^n = -\left(\frac{1}{2}\right)^n$$

Először nézzük meg, mi történik, ha $R = 1$. Ekkor ha felhasználjuk (2.15.) egyenletet, akkor:

$$\lim_{n\to\infty} \frac{|\delta_{n+1}|}{|\delta_n|^R} = \lim_{n\to\infty} \frac{\left|-\left(\frac{1}{2}\right)^{n+1}\right|}{\left|-\left(\frac{1}{2}\right)^{n}\right|} = \frac{1}{2} \neq 0$$

Vagyis mivel $R = 1$-re működik az algoritmus, és lineárisan konvergál. Innentől fogva mivel *R* értéke kielégíti a (2.15.) egyenletben megfogalmazott kritériumot, más értékeket nem is kell megvizsgálni.

3 Görbeillesztési eljárások [3]

A különböző valós mérnöki problémák során a legtöbb esetben valamilyen forrásból származó bejövő adatokat kell feldolgozni, kiértékelni, illetve ezek alapján megállapítani az adott rendszer, eszköz, modell viselkedését. Az egyik jellemző tulajdonsága ezen adatoknak, hogy leggyakrabban valamilyen mérés végrehajtása során generálódnak, vagyis valamilyen fizikai mennyiségek kvantitatív kiértékelése során tudunk hozzájutni. Így például egy áramló közegnek tudjuk mérni a sebességét, nyomását, hőmérsékletét, és ismerve anyagjellemzőit, illetve az rendelkezésre álló áramlási teret, megállapíthatjuk a tömegáramot. Hasonlóképpen egy áramkörben mérve a feszültséget és áramerősséget tudjuk számítani az áramkör ellenállását, vagy éppen egy ismert karakterisztikájú rugó kinyúlásából/összenyomódásából tudunk erőt mérni. Mindegyik esetben két dolog lesz szembeötlő: az egyik, hogy csak ritkán, szerencsés esetben tudjuk közvetlenül azokat a paramétereket mérni, amire tényleg szükségünk van; a másik, hogy mivel emberek végzik/tervezik a méréseket, így a hiba kódolva van benne (ezért is született meg a jól ismert „egy mérés nem mérés" szállóige).

Az viszont szinte biztos, hogy az egyik legjobb kiindulási pont, ha ezen adatokat azonnal ábrázoljuk egy célszerűen megválasztott diagramon, hogy lássuk, a vizsgált paraméterek milyen módon viselkednek; lineáris, parabolikus jellegűek, exponenciális vagy normál eloszlásúak, esetleg teljesen sztochasztikusak. És ha már ott vagyunk, hogy van egy diagramunk benne a mért adatokkal, akkor azok osztályozását végezhetjük tudományos alapon. Ezen fejezet célja annak bemutatása, hogy a bejövő adatokra, mért pontokra milyen módon tudunk valamilyen egyenest, görbét, vagyis függvényt illeszteni, vagy legalábbis hogyan tudjuk ezeket a legjobban leírni matematikailag.

Itt már fel is hívnám a figyelmet a fejezet egyik legfontosabb tanulságára: a rendelkezésre álló, derékszögű koordináta rendszerben ábrázolt pontokhoz kétféleképpen tudunk függvényt illeszteni:

- regresszióval
- interpolációval.

3.1 Regresszió

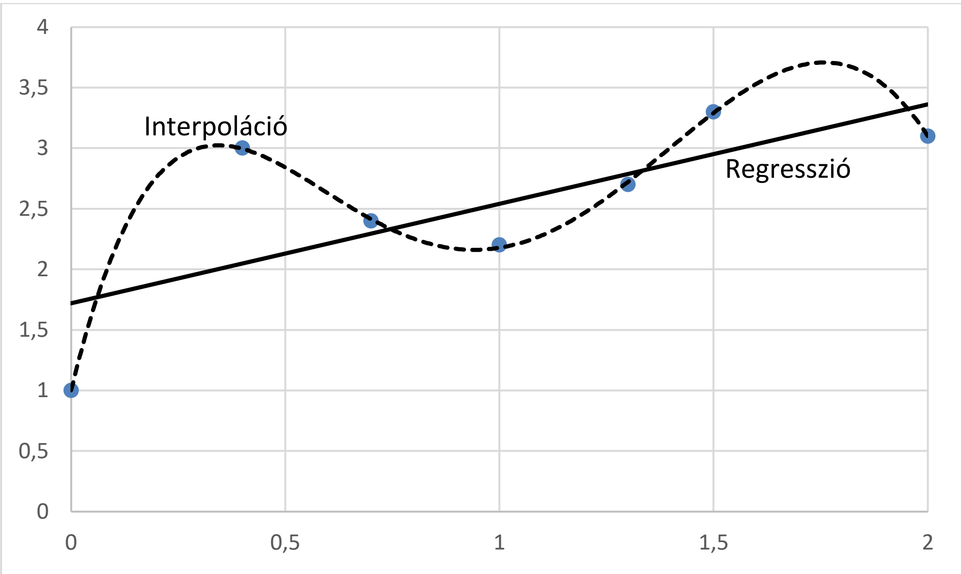
A regresszió kiötlése Francis Galton nevéhez fűződik (most már tudjuk ki áll a háttérben), és jelentése: visszatérés az átlaghoz. Lényegében véve a regresszió nem más, mint annak a vizsgálata, hogy az eredményváltozó mily módon függ az ún. magyarázó változók alakulásától. Sok ilyen példát lehetne felhozni:

- hogy hogyan alakul egy repülőtér forgalma az egy főre eső GDP függvényében,
- hogyan változik egy motor által leadott nyomaték a fordulatszám függvényében,
- hogy hogyan nő az esélye az elégtelentől különböző jegyek megszerzésének a tanult órák függvényében Mérnöki Számítások tárgyból.

Természetesen ezek csak kiragadott példák, mert a legtöbb esetben nem csak két változó szerepel egy problémában, így a repülőtér forgalma nem csak a GDP-től függ, befolyásolja az elhelyezkedés, éghajlat, turizmus, ..., a motor esetében befolyásoló tényezők a használt üzemanyag minősége, terhelés, ..., tanulásnál pedig hát... az tényleg csak egyváltozós feladat.

A cél, hogy regressziószámítással két vagy több véletlen változó közötti kapcsolatot próbáljuk meg modellezni. Alapvetően két fajta regresszió típust különböztetünk meg egymástól: a lineárist és a nemlineárist.

Igen fontos megérteni, hogy a regressziószámítás jellegéből adódóan, akár lineáris akár nemlineáris közelítést alkalmazunk, csak egy közelítés lesz, vagyis a célfüggvénynek nem kell átmennie az alappontokon. Természetbeni különbségüket a 2. ábra szemlélteti.



2. ábra: Ugyanazon alappontok lineáris regresszióval való közelítése és interpolációval való görbeillesztés alkalmazása közötti különbség bemutatása

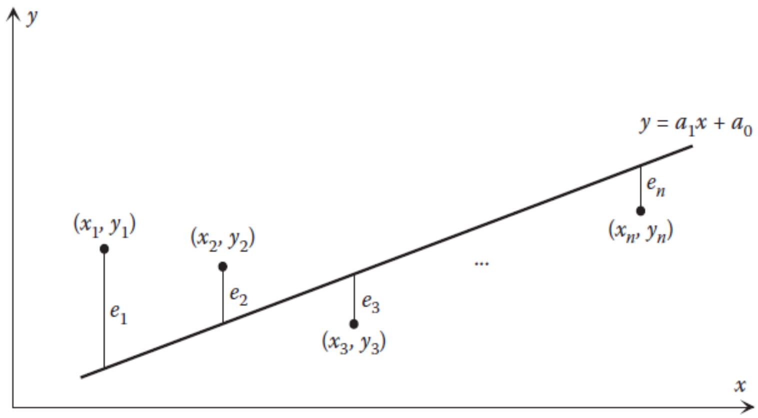
Általánosságban véve elmondható, hogy regressziót nagy számú vagy szórással rendelkező adathalmaz esetén, illetve amikor a bemenő adatok karakterisztikája előre ismert (lásd nyomatékgörbe) érdemes alkalmazni.

3.1.1 Lineáris regresszió

A regresszió legegyszerűbb formája a lineáris regresszió, vagyis amikor tudjuk, hogy az adatok lineárisan növekvő vagy csökkenő jelleget mutatnak. Ez a mérnöki alkalmazások során igen sűrűn előforduló probléma, mivel igen szeretünk lineáris rendszerekkel foglalkozni. A lineáris regresszió feladata, hogy megtalálja azon egyenes képletét

$$y = a_1 x + a_0 \tag{3.1.}$$

alakban, mely a legjobban illik az n darab alappontra: $(x_1, y_1), \dots, (x_n, y_n)$-re. Ehhez első körben érdemes megjeleníteni az adatokat egy derékszögű koordináta rendszerben, hogy szemrevételezzéssel is meggyőződjünk róla, hogy az adatok lineáris jelleget mutatnak. Ha ránézésre látszik például, hogy az adatok köbös jellegűek, a lineáris regresszió továbbra is működni fog, de az eredménye használhatatlan.



3. ábra: Lineáris adathalmaz a kezdeti hibákkal [3]

A feladatunk, hogy a_1 és a_0 értékét úgy határozzuk meg, hogy az egyenes és az alappontok közötti távolság (hiba) a lehető legkisebb legyen. Ezt a hibát a abszolút hibaként a következőképpen definiálhatjuk egy tetszőleges x_i pontban:

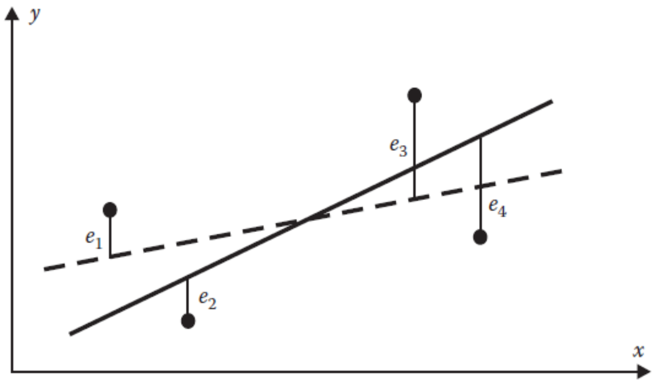
$$e_i = y_i - (a_1 x_i + a_0) \tag{3.2.}$$

Ezen egyes hibák kerülnek kiszámításra a teljes értelmezési tartományon a legjobb a_1 és a_0 értékek meghatározása érdekében.

De hogyan döntjük el, melyik a legjobb megoldás? Láthatjuk, hogy az abszolút hibák számításánál nem vettünk abszolút értéket, így ha ezen értékeket összeadogatjuk, a negatív és pozitív előjelű hibák kioltva egymást abba a hamis illúzióba engednek, hogy a megoldásunk pontos, pedig ez koránt sincs így. Ennek kiküszöbölése érdekében két dolgot tehetünk: vagy vesszük az abszolút értékét a hibáknak és úgy összegezzük őket, vagy pedig négyzetre emeljük őket és úgy végezzük ez a szummázást. Mindkét esetben a hibák egymást kioltó hatása megszűnik, és a végeredmény mindig pozitív szám lesz. Viszont az első esetben nem feltétlenül kapunk egyértelmű megoldást ugyanazon pontokra, mivel ha megoldjuk következő egyenletet:

$$E = \sum_{i=1}^{n} e_i = \sum_{i=1}^{n} |y_i - (a_1 x_i + a_0)| \qquad (3.3.)$$

akkor például a következő esetben (lásd 4. ábra) a 4 alappontra a szaggatott és folytonos vonallal jelölt egyenesek is megoldást jelentenek.



4. ábra: Két lineáris illesztés ugyanakkora össz-abszolút hibával

Ennek kiküszöbölésére terjedt el a leginkább a hibák négyzetének összegzése, mely igen meglepő módon a legkisebb négyzetek módszere nevet kapta:

$$E = \sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} [y_i - (a_1 x_i + a_0)]^2 \qquad (3.4.)$$

Ez a módszer az összes eddigi előnyt magába foglalja:

- a különböző előjelű hibák nem ütik ki egymást,
- az összhiba mindig pozitív lesz,
- egyértelmű megoldás található a_1 és a_0 értékére,
- kis hibák négyzete még kisebb lesz, így kisebb súllyal szerepelnek, mint a nagy hibák, melyeknek négyzete pedig súlyosabb, így sokkal pontos közelítés kivitelezhető.

A lineáris legkisebb négyzetek módszerének megoldása:

A cél továbbra is egy olyan $y = a_1x + a_0$ egyenlet keresése, mely a legközelebb fekszik $(x_1, y_1), \dots, (x_n, y_n)$ alappontokra, és minimalizálja a (3.4.) egyenletben leírt összhibát. Tekintetbe véve, hogy E összhiba a_1-től és a_0-tól függ, méghozzá nemlineáris módon, így adódik, hogy ott van minimuma, ahol $\frac{\partial E}{\partial a_1}$ és $\frac{\partial E}{\partial a_0}$ értéke nulla lesz, vagyis:

$$\frac{\partial E}{\partial a_1} = -2\sum_{i=1}^{n} x_i[y_i - (a_1x_i + a_0)] = 0$$
$$\to \sum_{i=1}^{n}\{x_i[y_i - (a_1x_i + a_0)]\} = 0 \qquad (3.5.)$$
$$\frac{\partial E}{\partial a_0} = -2\sum_{i=1}^{n}[y_i - (a_1x_i + a_0)] = 0 \to \sum_{i=1}^{n}[y_i - (a_1x_i + a_0)] = 0$$

E két egyenletet kifejtve és átrendezve a következő egyenletekhez jutunk, melyeket megoldva megkaphatjuk a_1 és a_0 értékét:

$$\left(\sum_{i=1}^{n} x_i\right)a_0 + \left(\sum_{i=1}^{n} x_i^2\right)a_1 = \sum_{i=1}^{n} x_iy_i$$
$$na_0 + \left(\sum_{i=1}^{n} x_i\right)a_1 = \sum_{i=1}^{n} y_i \qquad (3.6.)$$

Bevetve a Cramer szabályt a megoldás pedig:

$$a_0 = \frac{\left(\sum_{i=1}^{n} x_i^2\right)\left(\sum_{i=1}^{n} y_i\right) - \left(\sum_{i=1}^{n} x_i\right)\left(\sum_{i=1}^{n} x_iy_i\right)}{n\left(\sum_{i=1}^{n} x_i^2\right) - \left(\sum_{i=1}^{n} x_i\right)^2}$$
$$a_1 = \frac{n\left(\sum_{i=1}^{n} x_iy_i\right) - \left(\sum_{i=1}^{n} x_i\right)\left(\sum_{i=1}^{n} y_i\right)}{n\left(\sum_{i=1}^{n} x_i^2\right) - \left(\sum_{i=1}^{n} x_i\right)^2} \qquad (3.7.)$$

Habár a (3.7.) egyenletek elég rettenetesen néznek ki, ez csak azért van, mert nem kapásból Excellel gondolkodunk. Nézzünk rá egy példát:

3.1. Példa

Feladat:

Elektrotechnika laboratóriumi mérésünk során azt tapasztaljuk, hogy egy elektromos fogyasztó ellenállása a következőképpen alakul annak hőmérséklete függvényében:

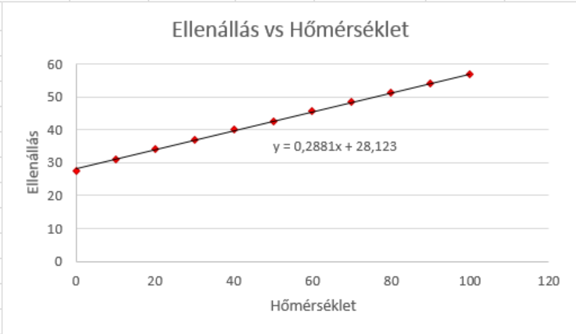
x [°C]	0	10,0	20,0	30,0	40,0	50,0	60,0	70,0	80,0	90,0	100,0
y [Ω]	27,6	31,0	34,0	37,0	40,0	42,6	45,5	48,3	51,1	54,0	56,7

Természetesen mindezt azért tettük, hogy demonstráljuk azt a tényt, hogy az ellenállás növekszik a hőmérséklet függvényében. De nem is ez a lényeg, hanem hogy keressük meg ennek az adathalmaznak a lineáris egyenesét, mely a legkisebb hibát adja!

Megoldás:

Hozzuk hozzá létre a következő táblázatot, ahol a már jól ismert képleteket alkalmazzuk:

	A	B	C	D	E	F	G	H	I	J	K	L
1	**Lineáris Regresszió példa**											
2	x [°C]	y [Ω]	x^2	xy								
3	0	27,6	0	0								
4	10,0	31,0	100,0	310,0								
5	20,0	34,0	400,0	680,0								
6	30,0	37,0	900,0	1110,0								
7	40,0	40,0	1600,0	1600,0								
8	50,0	42,6	2500,0	2130,0								
9	60,0	45,5	3600,0	2730,0								
10	70,0	48,3	4900,0	3381,0								
11	80,0	51,1	6400,0	4088,0								
12	90,0	54,0	8100,0	4860,0								
13	100,0	56,7	10000,0	5670,0								
14	**550,0**	**467,8**	**38500,0**	**26559,0**								
15												
16	$\sum x^2$	$\sum x$		38500	550	Δ						
17	$\sum x$	n	=	550	11	=	121000		121000			
18												
19								**$a_1 = D_1/\Delta =$ 0,288091**				
20	$\sum xy$	$\sum x$		26559,0	550,0	D_1						
21	$\sum y$	n	=	467,8	11	=	34859		Megoldás: $y = a_1x + a_0 =$		0,288091x + 28,12273	
22												
23	$\sum x^2$	$\sum xy$		38500,0	26559,0	D_2		**$a_0 = D_2/\Delta =$ 28,12273**				
24	$\sum x$	$\sum y$	=	550,0	467,8	=	3402850					



Természetesen a cél az lenne, hogy a szürke cellákba ne kézzel írjuk be a számokat (azt bármelyik félkegyelmű tudja), hanem hogy ezen a helyeken képletek dolgozzanak helyettünk. Ne felejtsük el alkalmazni G17, G21 és G24 cellákban a mátrix determinánsát számító függvényünket!

Ha illesztünk egy egyenest a pontokra az Excel beépített görbeillesztőjével, akkor csodák csodájára ugyanazt fogjuk visszakapni, mint amit mi is számítottunk, ezzel is igazolva önmagunkat.

3.2. Példa

Feladat:

Az előzőhöz hasonlóan továbbra is az Elektrotechnika laborban sertepertélve a következő feszültség – áramerősség adatokat sikerült kimérni:

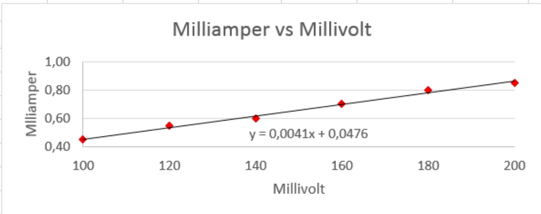
x [mV]	100	120	140	160	180	200
y [mA]	0,45	0,6	0,6	0,7	0,8	0,9

Csak úgy, mint az előbb, hajtsunk végre lineáris regressziót!

Megoldás:

Ugyanazon analógiával, mint az előbb, hozzuk létre a következő táblázatot:

	A	B	C	D	E	F	G	H	I	J	K	L
1	**Lineáris Regresszió példa**											
2	x [mV]	y [mA]	x^2	xy								
3	100	0,45	10000	45								
4	120,0	0,55	14400,0	66,0								
5	140,0	0,60	19600,0	84,0								
6	160,0	0,70	25600,0	112,0								
7	180,0	0,80	32400,0	144,0								
8	200,0	0,85	40000,0	170,0								
9	**900,0**	**3,95**	**142000,0**	**621,0**								
10												
11	$\sum x^2$	$\sum x$		142000	900	Δ						
12	$\sum x$	n	=	900	6	=	42000					
13												
14								**$a_1 = D_1/\Delta =$**	**0,00407**			
15	$\sum xy$	$\sum x$		621,0	900,0	D_1						
16	$\sum y$	n	=	4,0	6	=	171		Megoldás: $y = a_1x + a_0 =$	0,004071x + 0,047619		
17												
18	$\sum x^2$	$\sum xy$		142000,0	621,0	D_2		**$a_0 = D_2/\Delta =$**	**0,04762**			
19	$\sum x$	$\sum y$	=	900,0	4,0	=	2000					



Szintén ellenőrzésképpen érdemes illeszteni egy görbét, megnézni sikerült-e a számításunk.

3.3. Példa

Feladat:

A lineáris regresszió mellesleg nem csak pontokra való egyenes illesztésére alkalmas, de akár túlhatározott egyenletrendszer megoldására is bevethető. Tegyük fel, hogy 6 ember 6 különböző lineáris egyenletre esküszik, és nekünk kell megtalálni az arany középutat. A 6 egyenlet a következő:

$$2x + y = -1$$

$$x - 3y = -4$$
$$x + 4y = 3$$
$$3x - 2y = -6$$
$$-x + 2y = 3$$
$$x + 3y = 2$$

Megoldás:

Feladatunk bonyolultnak tűnik, de igazából pont olyan könnyű, mint eddig volt. Ahhoz, hogy ezt belássuk, nem kell mást tenni, mert a két változó együtthatóit kigyűjteni egy táblázatba a konstansokkal együtt, és a következő módon (táblázat szerint) végrehajtani a lineáris regressziót, hogy visszaállhasson a béke a galaxisban:

	A	B	C	D	E	F	G	H	I	J	K
1	**Túlhatározott lineáris egyenletrendszer megoldása lineáris regressszióval**										
2		**a**	**b**	**c**	**a^2**	**ab**	**b^2**	**ac**	**bc**		
3		2	1	-1	4	2	1	-2	-1		
4		1	-3	-4	1	-3	9	-4	12		
5		1	4	3	1	4	16	3	12		
6		3	-2	-6	9	-6	4	-18	12		
7		-1	2	3	1	-2	4	-3	6		
8		1	3	2	1	3	9	2	6		
9											
10	∑	7	5	-3	17	-2	43	-22	47		
11											
12	Δ	$\sum a^2$	$\sum ab$	=	17	-2	=	727		x együtthatója:	
13		$\sum ab$	$\sum b^2$		-2	43	Δ			**$x = D_1/\Delta =$**	-1,17194
14											
15	D_1	$\sum ac$	$\sum ab$	=	-22	-2	=	-852			
16		$\sum bc$	$\sum b^2$		47	43	D_1			y együtthatója:	
17										**$y = D_2/\Delta =$**	1,03851
18	D_2	$\sum a^2$	$\sum ac$	=	17	-22	=	755			
19		$\sum ab$	$\sum bc$		-2	47	D_2				

A két megkapott érték (ez esetbe *x* és *y*) pedig az összes egyenlethez legközelebbi egyenlet együtthatóit adja ki.

3.1.2 Nemlineáris adatok linearizálása

Ha az adatok között kapcsolat nem lineáris, akkor kétféleképpen járhatunk el: vagy polinomiális regressziót hajtunk végre (lásd következő fejezet) vagy az adatokat valamilyen módon lineárissá varázsoljuk át, melyet aztán felhasználhatunk egy lineáris regresszióhoz. A nemlineáris adatok linearizálásához alapvetően háromféle függvényt használhatunk: exponenciális függvényt, hatvány függvényt és hányados függvényt. A következő alfejezet ezt a három technikát mutatja be.

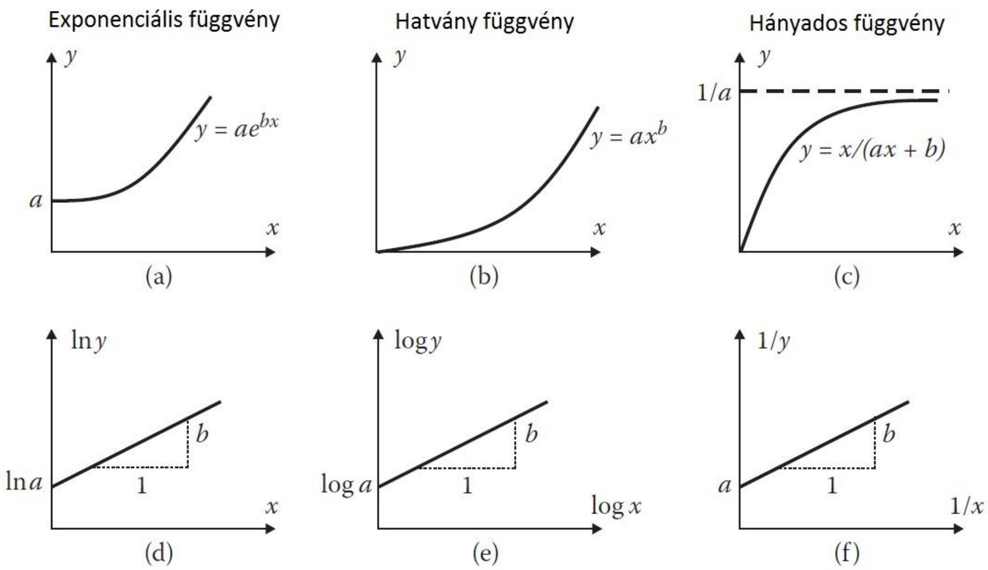
3.1.2.1 Exponenciális függvény

Az exponenciális függvény természetesen a következő alakban írható fel:

$$y = ae^{bx} \tag{3.8.}$$

ahol a és b konstansok.

Mivel az exponenciális függvény deriváltja a függvény konstans kitevőjének szorzatával vett függvényhez tér vissza, így ez a technika elsősorban olyan mennyiségek változásának leírására használható, melynek megváltozásának aránya egyenesen arányos a mennyiséggel magával, mint például a radioaktív bomlás, vagy éppen a lángterjedési sebesség Vibe modellel leírva.



5. ábra: A három linearizáló függvény görbeillesztéshez: a, d – exponenciális; b, e – hatvány; c, f – hányados

A linearizáláshoz nem kell mást tenni, mint a (3.8.) egyenlet természetes alapú logaritmusát venni a következő módon:

$$\ln y = bx + \ln a \tag{3.9.}$$

Így az $\ln y$ ábrázolása x függvényében egy egyenest fog adni b meredekséggel és $\ln a$ metszési ponttal, ahogy az 5. ábra *a* és *d* diagramja mutatja.

3.1.2.2 Hatvány függvény

A második függvényünk a hatvány függvény, melynek alakja nem szabad, hogy meglepetést okozzon:

$$y = ax^b \tag{3.10.}$$

ahol a és b szintén konstansok. A linearizáláshoz az egyenlet 10-es alapú logaritmusát vesszük:

$$\log y = b \log x + \log a \tag{3.11.}$$

vagyis $\log y$ értékét $\log x$ függvényében kell ábrázolnunk, hogy egy b meredekségű, $\log a$ metszéspontú egyenest kapjunk, ahogy 5. ábra b és e diagramja is mutatja.

3.1.2.3 Hányados függvény

A hányados függvény pedig végül a következő alakban írható fel:

$$y = \frac{x}{ax + b} \tag{3.12.}$$

Szintén a és b konstans értékek.

Ezt a függvényt a következőképpen alakítjuk át reciprokát véve:

$$\frac{1}{y} = b\left(\frac{1}{x}\right) + a \tag{3.13.}$$

vagyis az $1/y$ megjelenítése $1/x$ függvényében történik ahhoz, hogy b meredekségű és a metszéspontú egyenest kapjunk, ahogy az 5. ábra *c* és f diagramja szemlélteti.

3.1.3 Polinomiális regresszió

Természetesen nagyon sok olyan eset van, amikor a lineáris regresszió nem működik, mivel a bemenő adatok egyértelműen parabolikus, köbös... jelleget mutatnak. Ezekben az esetekben érdemes alkalmazni a polinomiális regressziót. Ha a 3.1.1 fejezetet kiterjesztjük *n* darab pontra: $(x_1, y_1), \dots, (x_n, y_n)$, melyekre egy *m*-ed fokú polinomot szeretnénk illeszteni a következő alakban:

$$y = a_m x^m + a_{m-1} x^{m-1} + \dots + a_2 x^2 + a_1 x + a_0 \tag{3.14.}$$

akkor az összhiba a következő formában írható fel:

$$E = \sum_{i=1}^{n} [y_i - (a_m x^m + a_{m-1} x^{m-1} + \cdots + a_2 x^2 + a_1 x + a_0)]^2 \qquad (3.15.)$$

Az együtthatók $(a_m, \dots, a_1, a_0)$ hasonlóan határozhatók meg, mint a lineáris esetben, vagyis egyesével deriválnunk kell velük E értékét, a deriváltakat pedig nullával egyenlővé tenni, hogy a minimumhelyeket meghatározhassuk:

$$\begin{aligned}
\frac{\partial E}{\partial a_0} &= -2 \sum_{i=1}^{n} [y_i - (a_m x^m + a_{m-1} x^{m-1} + \cdots + a_2 x^2 + a_1 x + a_0)] = 0 \\
\frac{\partial E}{\partial a_1} &= -2 \sum_{i=1}^{n} \{x_i [y_i - (a_m x^m + a_{m-1} x^{m-1} + \cdots + a_2 x^2 + a_1 x + a_0)]\} = 0 \\
\frac{\partial E}{\partial a_2} &= -2 \sum_{i=1}^{n} \{x_i^2 [y_i - (a_m x^m + a_{m-1} x^{m-1} + \cdots + a_2 x^2 + a_1 x + a_0)]\} = 0 \\
&\dots \\
\frac{\partial E}{\partial a_m} &= -2 \sum_{i=1}^{n} \{x_i^m [y_i - (a_m x^m + a_{m-1} x^{m-1} + \cdots + a_2 x^2 + a_1 x + a_0)]\} = 0
\end{aligned} \qquad (3.16.)$$

Ezt a roppant egyenletrendszer szintén átalakítható $m + 1$ lineáris egyenletté, melyek így megoldhatók lesznek $a_m, \dots, a_2, a_1, a_0$ együtthatókra:

$$\begin{aligned}
n a_0 + \left(\sum_{i=1}^{n} x_i\right) a_1 + \left(\sum_{i=1}^{n} x_i^2\right) a_2 + \cdots + \left(\sum_{i=1}^{n} x_i^m\right) a_m &= \sum_{i=1}^{n} y_i \\
\left(\sum_{i=1}^{n} x_i\right) a_0 + \left(\sum_{i=1}^{n} x_i^2\right) a_1 + \left(\sum_{i=1}^{n} x_i^3\right) a_2 + \cdots + \left(\sum_{i=1}^{n} x_i^{m+1}\right) a_m &= \sum_{i=1}^{n} x_i y_i \\
\left(\sum_{i=1}^{n} x_i^2\right) a_0 + \left(\sum_{i=1}^{n} x_i^3\right) a_1 + \left(\sum_{i=1}^{n} x_i^4\right) a_2 + \cdots + \left(\sum_{i=1}^{n} x_i^{m+2}\right) a_m &= \sum_{i=1}^{n} x_i^2 y_i \\
&\dots \\
\left(\sum_{i=1}^{n} x_i^m\right) a_0 + \left(\sum_{i=1}^{n} x_i^{m+1}\right) a_1 + \left(\sum_{i=1}^{n} x_i^{m+2}\right) a_2 + \cdots + \left(\sum_{i=1}^{n} x_i^{2m}\right) a_m &= \sum_{i=1}^{n} x_i^m y_i
\end{aligned} \qquad (3.17.)$$

Persze a gyakorlatban m értéke ritkán megy 2 vagy 3 fölé, de (3.17.) egyenletek tulajdonképpen tetszőleges m értékére megoldást szolgáltatnak.

3.4. Példa

Feladat:

Adott a következő adatsor:

x	1,20	1,50	1,80	2,60	3,10	4,30	4,90	5,30
y	4,50	5,10	5,80	6,70	7,00	7,30	7,60	7,40
x	5,70	6,40	7,10	7,60	8,60	9,20	9,80	
y	7,20	6,90	6,60	5,10	4,50	3,40	2,70	

Illesszünk rá parabolikus (másodfokú) polinomot!

Megoldás:

A másodfokú polinom egyenlete:

$$y = a_2x^2 + a_1x + a_0$$

Így az összhiba:

$$E = \sum_{i=1}^{n} [y_i - (a_2x^2 + a_1x_i + a_0)]^2$$

Ennek a minimalizálását kell elvégeznünk. Mivel itt $m = 2$, így (3.17.)-ből szerencsére csak 3 egyenlet marad meg:

$$na_0 + \left(\sum_{i=1}^{n} x_i\right)a_1 + \left(\sum_{i=1}^{n} x_i^2\right)a_2 = \sum_{i=1}^{n} y_i$$

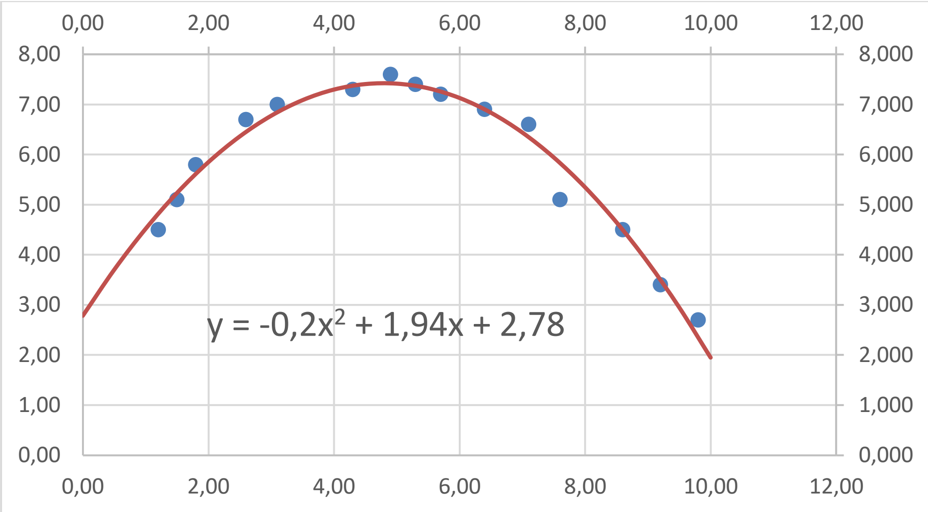
$$\left(\sum_{i=1}^{n} x_i\right)a_0 + \left(\sum_{i=1}^{n} x_i^2\right)a_1 + \left(\sum_{i=1}^{n} x_i^3\right)a_2 = \sum_{i=1}^{n} x_iy_i$$

$$\left(\sum_{i=1}^{n} x_i^2\right)a_0 + \left(\sum_{i=1}^{n} x_i^3\right)a_1 + \left(\sum_{i=1}^{n} x_i^4\right)a_2 = \sum_{i=1}^{n} x_i^2y_i$$

Így pedig egy Excel táblában a következő formában tudjuk megoldani a problémát:

	A	B	C	D	E	F	G
1	**Parabolikus regresszió**						
2	x	y	x^2	x^3	x^4	xy	x^2y
3	1,20	4,50	1,44	1,73	2,07	5,40	6,48
4	1,50	5,10	2,25	3,38	5,06	7,65	11,48
5	1,80	5,80	3,24	5,83	10,50	10,44	18,79
6	2,60	6,70	6,76	17,58	45,70	17,42	45,29
7	3,10	7,00	9,61	29,79	92,35	21,70	67,27
8	4,30	7,30	18,49	79,51	341,88	31,39	134,98
9	4,90	7,60	24,01	117,65	576,48	37,24	182,48
10	5,30	7,40	28,09	148,88	789,05	39,22	207,87
11	5,70	7,20	32,49	185,19	1055,60	41,04	233,93
12	6,40	6,90	40,96	262,14	1677,72	44,16	282,62
13	7,10	6,60	50,41	357,91	2541,17	46,86	332,71
14	7,60	5,10	57,76	438,98	3336,22	38,76	294,58
15	8,60	4,50	73,96	636,06	5470,08	38,70	332,82
16	9,20	3,40	84,64	778,69	7163,93	31,28	287,78
17	9,80	2,70	96,04	941,19	9223,68	26,46	259,31
18	$\sum x=$	$\sum y=$	$\sum x^2=$	$\sum x^3=$	$\sum x^4=$	$\sum xy=$	$\sum x^2y=$
19	79,10	87,80	530,15	4004,50	32331,49	437,72	2698,37
20							
21		530,15	79,10	15,00		$\sum y=$	87,80
22	A=	4004,50	530,15	79,10		$\sum xy=$	437,72
23		32331,49	4004,50	530,15		$\sum x^2y=$	2698,37
24							
25		0,032	-0,016	0,002		$a_2=$	-0,20
26	$A^{-1}=$	-0,385	0,181	-0,016		$a_1=$	1,94
27		0,978	-0,385	0,032		$a_0=$	2,78
28							
29						$y = a_2x^2 + a_1x + a_0$	

Ezekkel az együtthatókkal a következő módon fogja közelíteni a kiindulási pontokat a parabolánk:



Ugyanilyen módon kivitelezhető a köbös, vagy magasabb rendű regresszió is.

3.2 Véges differenciák

A véges különbségek a numerikus analízisek egyik elengedhetetlen, legfontosabb alapját képezik, így erre fog épülni a görbeillesztés, adatok simítása, későbbiekben pedig numerikus differenciálás és integrálás. Nézzük meg elméleti oldalról, hogy miről is van szó.

Tegyük fel, hogy van egy folytonos $y = f(x)$ függvényünk, és adott $x_0, x_1, x_2, \dots, x_n$ értékek az x értelmezési tartományon, melyekre igaz, hogy $x_0 \leq x \leq x_n$. Ebből kiindulva az ezen a tartományon értelmezett $f(x)$ függvény a következő általános jelölésű értékekkel rendelkezik:

x	$\mathbf{f(x)}$
x_0	$f(x_0)$
x_1	$f(x_1)$
x_2	$f(x_2)$
…	…
x_{n-1}	$f(x_{n-1})$
x_n	$f(x_n)$

1. táblázat: $x_0 \leq x \leq x_n$ tartományon vett függvény $f(x)$ értékek általános jelölése

Legyen x_i és x_j két, nem feltétlenül egymást követő értékei az x intervallumból. Ekkor, az első osztott differenciát a következőképpen írhatjuk fel:

$$f(x_i, x_j) = \frac{f(x_i) - f(x_j)}{x_i - x_j} \tag{3.18.}$$

Hasonlóan a második osztott differencia:

$$f(x_i, x_j, x_k) = \frac{f(x_i, x_j) - f(x_j, x_k)}{x_i - x_k} \tag{3.19.}$$

Hasonlóan definiálhatnánk a harmadik, negyedik... differenciákat is.

Az osztott differenciákat általában egy differencia táblázatban jelenítjük meg, ahol minden differencia két előző érték közé kerül beírásra, ahogy a következő táblázat is szemlélteti ezt:

x	$f(x)$			
x_0	$f(x_0)$			
		$f(x_0, x_1)$		
x_1	$f(x_1)$		$f(x_0, x_1, x_2)$	
		$f(x_1, x_2)$		$f(x_0, x_1, x_2, x_3)$
x_2	$f(x_2)$		$f(x_1, x_2, x_3)$	
		$f(x_2, x_3)$		
x_3	$f(x_3)$			

2. táblázat: Osztott differenciák felírása $f(x)$ függvényre

Mivel eddig ez elég ködös lehet, nézzünk rá egy példát!

3.5. Példa

Feladat

Adjuk meg a differencia táblázatát a következő x értékekre: 0, 1, 3, 4, 7, 9 az alábbi függvénynek: $y = f(x) = x^3$ és az elsőtől a 4 osztott differenciáit.

Megoldás

Létrehozzuk a lenti táblázatot. Az első oszlop tartalmazza az x értékeit, a második oszlop az f(x) függvény értékeket, a harmadiktól a hatodikig pedig a differenciákat. Az előbb már leírt képlet segítségével a következőképpen számíthatjuk pl. az első osztott differenciát:

$$\frac{1-27}{1-3}=13$$

a második osztott differencia pedig:

$$\frac{37-93}{3-7}=14$$

És így tovább, a harmadik differencia:

$$\frac{4-8}{0-4}=1$$

Végül pedig a negyedik:

$$\frac{1-1}{0-4}=0$$

És íme a táblázat (érdekes megfigyelni, hogy a 3. osztott differenciák már azonosak, míg a 4. már ki is nullázódik):

Függvény		**Osztott differencia**			
x	$f(x) = x^3$	**Első**	**Második**	**Harmadik**	**Negyedik**
0	**0**				
		1			
1	**1**		4		
		13		1	
3	**27**		8		0
		37		1	
4	**64**		14		0
		93		1	
7	**343**		20		
		193			
9	**729**				

Nézzük meg, hogyan lehetne általánosítani ezt a módszert.

Általában véve ezt az x értékek egyenlő távolságra vannak egymástól vagyis ekvidisztáns kiosztással rendelkeznek. Ebben az esetben a differenciák szintén egymást követő értékek. Ráadásul a nevezők is azonosak lesznek, így azokat ki lehet ejteni. Ebben az esetben a differenciák csak a függvény különbségeiként állnak elő.

Szóval, ha a különbség állandó az x értékei között, akkor egy teljesen átlagos x_k érték felírható a következőképpen:

$$x_k = x_0 + kh \quad ahol\ k = \cdots -2, -1, 0, 1, 2, \ldots \tag{3.20.}$$

Ez alapján kifejezhetjük az első differenciákat, ha bevezetjük Δ különbség operátort:

$$\Delta f_k = f_{k+1} - f_k \tag{3.21.}$$

Ehhez hasonlóan a második differenciák:

$$\Delta^2 f_k = \Delta(\Delta f_k) = \Delta f_{k+1} - \Delta f_k \tag{3.22.}$$

és általánosságban pozitív egész n értékekre:

$$\Delta^n f_k = \Delta(\Delta^{n-1} f_k) = \Delta^{n-1} f_{k+1} - \Delta^{n-1} f_k \tag{3.23.}$$

A különbség operátor persze megfelel a kitevők szabályának, vagyis:

$$\Delta^m(\Delta^n f_k) = \Delta^{m+n} f_k \tag{3.24.}$$

Ez alapján az új differencia táblázat a következőképpen írható fel:

Függvény		Osztott differencia			
x	$f(x) = x^3$	**Első**	**Második**	**Harmadik**	**Negyedik**
x_0	f_0				
		Δf_0			
x_1	f_1		$\Delta^2 f_0$		
		Δf_1		$\Delta^3 f_0$	
x_2	f_2		$\Delta^2 f_1$		$\Delta^4 f_0$
		Δf_2		$\Delta^3 f_1$	
x_3	f_3		$\Delta^2 f_2$		
		Δf_3			
x_4	f_4				
…					
x_n	f_n				

3. táblázat: Differencia táblázat általános felírása

3.6. Példa

Feladat:

Adottak a következő x értékek: 1,2,3,4,5,6,7,8.

Ezeken az előző jóbarát $f(x) = x^3$ függvényt értelmezzük, és keressük a differenciákat!

Megoldás:

Megalkotva a differencia táblázatot a már ismert módon konstatálhatjuk, hogy a 3. differenciák ismét egyenlők, a 4. differenciák pedig 0-k lesznek.

Függvény			Differenciák				
k	x_k	f_k	Δf_k	$\Delta^2 f_k$	$\Delta^3 f_k$	$\Delta^4 f_k$	…
1	1	1					
			7				
2	2	8		12			
			19		6		
3	3	27		18		0	
			37		6		
4	4	64		24		0	
			61		6		
5	5	125		30		0	
			91		6		
6	6	216		36		0	
			127		6		
7	7	343		42			
			169				
8	8	512					

Amennyiben binomiális eloszlást nézünk:

$$\binom{n}{j} = f_{k+1} - f_k \tag{3.25.}$$

Ekkor a (3.23.) képlet módosítva a következő formában írható fel:

$$\Delta^n f_k = f_{k+n} - n f_{k+n-1} + \frac{n(n-1)}{2!} f_{k+n-2} + \cdots + (-1)^{n-1} n f_{k+1} + (-1)^n f_k \tag{3.26.}$$

Ez felírva $k = 0, n = 1,2,3,4$ értékekre barátságosabban néz ki:

$$\begin{aligned} \Delta f_0 &= f_2 - f_1 \\ \Delta^2 f_0 &= f_2 - 2f_1 + f_0 \\ \Delta^3 f_0 &= f_3 - 3f_2 + 3f_1 - f_0 \\ \Delta^4 f_0 &= f_4 - 4f_3 + 6f_2 - 4f_1 + f_0 \end{aligned} \tag{3.27.}$$

3.2.1 Faktoriális polinomok

A faktoriális polinomok a következőképpen definiálhatók:

$$\begin{aligned} (x)^{(n)} &= x(x-1)(x-2)\ldots(x-n+1) \\ (x)^{-(n)} &= \frac{1}{(x-1)(x-2)\ldots(x-n)} \end{aligned} \tag{3.28.}$$

Ha ismét használjuk Δ különbség operátort, akkor:

$$\Delta(x)^{(n)} = (x)^{(n-1)} \tag{3.29.}$$

$$\Delta(x)^{-(n)} = -n(x)^{(n-1)}$$

Előfordul, hogy egy polinomot érdemes faktoriális polinomként kifejezni. Pont ezért is foglalkozunk most ezzel. Az átalakításhoz a Maclaurin sorba-fejtést kell alkalmaznunk:

$$p_n(x) = a_0 + a_1(x)^{(1)} + a_2(x)^{(2)} + \cdots + a_n(x)^{(n)} \qquad (3.30.)$$

ahol is az a_k együtthatókat kell meghatározni. Ehhez az első összefüggésünk $x = 0$-ra adódik:

$$a_0 = p_n(0) \qquad (3.31.)$$

Az a_1 együttható kiszámításához (3.30.) egyenletből $p_n(x)$ növekményét határozzuk meg:

$$\Delta p_n(x) = 1x^0 a_1 + 2a_2(x)^{(1)} + 3a_3(x)^{(2)} + \cdots + na_n(x)^{(n-1)} \qquad (3.32.)$$

majd x-et egyenlővé tesszük 0-val, aminek következtében:

$$a_1 = \Delta p_n(0) \qquad (3.33.)$$

A következő differenciálással:

$$\Delta^2 p_n(x) = 2 \cdot 1a_2 + 3 \cdot 2a_3(x)^{(1)} + \cdots + n(n-1)a_n(x)^{(n-2)} \qquad (3.34.)$$

Az $x = 0$ miatt:

$$a_2 = \frac{\Delta^2 p_n(0)}{2 \cdot 1} = \frac{\Delta^2 p_n(0)}{2!} \qquad (3.35.)$$

Általában véve pedig:

$$a_j = \frac{\Delta^j p_n(0)}{j!} \quad \forall\, j = 0,1,2, \dots, n \qquad (3.36.)$$

Ekkor már joggal merülhet fel a kérdés, hogy akkor ez miért is jó nekünk? A válasz igen egyszerű: mivel ez egy differencia táblázat létrehozásának az egyik legegyszerűbb módszere.

Ahhoz, hogy ezt kivitelezhessük, 3 lépés végrehajtására van szükség:

1. A Maclaurin sorral kifejezett $p_n(x)$ értékét elosztjuk x-szel, melyből kapunk egy $q_0(x)$ hányadost és r_0 maradékot, mely az a_0 konstans értéke lesz. Így:
$$p_n(x) = r_0 + xq_0(x) \qquad (3.37.)$$
2. Elosztjuk $q_0(x)$-et $(x-1)$-gyel, hogy kifejezzük $q_1(x)$-et, és r_1 maradékot, mely a_1 konstanssal lesz egyenlő:
$$q_0(x) = r_1 + (x-1)q_1(x) \qquad (3.38.)$$

Ezt a kifejezést visszahelyettesítve a (3.37.)-ba, plusz az eredeti (3.30.) képletet felhasználva:

$$\begin{aligned} p_n(x) &= r_0 + x[r_1 + (x-1)q_1(x)] \\ &= r_0 + r_1(x)^{(1)} + x(x-1)q_1(x) \end{aligned} \tag{3.39.}$$

3. Elosztjuk $q_1(x)$-et $(x-2)$-vel, hogy megkapjuk $q_2(x)$-et és r_2 maradékot, mely most a_2-vel lesz egyenlő, így:

$$q_1(x) = r_2 + (x-2)q_2(x) \tag{3.40.}$$

Az előző egyenletbe visszahelyettesítve:

$$\begin{aligned} &p_n(x) = r_0 + r_1(x)^{(1)} + x(x-1)[r_2 + (x-2)q_2(x)] = \\ &= r_0 + r_1(x)^{(1)} + r_2(x)^{(2)} + x(x-1)(x-2)q_2(x) \end{aligned} \tag{3.41.}$$

Ezt a folyamatot folytathatjuk egészen addig, amíg a hányados kitevője eggyel alacsonyabb nem lesz az előző hányados kitevőjével, vagyis $(n+1)$-szer tudjuk elvégezni.

Ezzel megkapjuk az általános faktoriális polinom alakját:

$$p_n(x) = r_0 + r_1(x)^{(1)} + r_2(x)^{(2)} + \cdots + r_{n-1}(x)^{(n-1)} + r_n(x)^{(n)} \tag{3.42.}$$

ahol:

$$r_j = a_j = \frac{\Delta^j p_n(0)}{j!} \quad \forall\, j = 0,1,2,\dots,n \tag{3.43.}$$

Másképpen megfogalmazva:

$$\Delta^j p_n(0) = j!\, r_j \tag{3.44.}$$

3.2.2 „Anti-differenciálás”

Amennyiben egy függvénynek csak az első deriváltját ismerjük, a függvény megkapásához a differenciálás ellentétjét, „anti-differenciálást” kell végeznünk, amit más terminológiával akár integrálásnak is szoktunk hívni bizonyos körökben. De hogy miért is kell ilyen körmönfontan beszélni az integrálásról? Azért, mivel most tulajdonképpen az integrálási műveletek nélkül, differenciálszámítással leszünk képesek ugyanazt a problémát megoldani. Visszanyúlva a (3.29.) képlethez, az anti-differenciálás a következő módon adható meg:

$$\Delta^{-1}(x)^{(n)} = \frac{(x)^{(n+1)}}{(n+1)} \tag{3.45.}$$

Nézzük meg egy példán keresztül miért is olyan ismerős ez a képlet!

3.7. Példa

Feladat:

Anti-differenciáljuk a következő polinomot:

$$p(x) = x^4 - 5x^3 + 3x + 4$$

Megoldás:

Az előzőleg leírt lépések alapján a faktoriális polinomunk a következő lesz: (osztás, hányados és maradékképzés, ismétlés):

$$p_n(x) = 4 - (x)^{(1)} - 8(x)^{(2)} + (x)^{(3)} + (x)^{(4)}$$

Majd meghatározzuk a differencia táblázatot:

$$\Delta^0 p(0) = 0! \cdot 4 = 4$$
$$\Delta^1 p(0) = 1! \cdot (-1) = -1$$
$$\Delta^2 p(0) = 2! \cdot (-8) = -16$$
$$\Delta^3 p(0) = 3! \cdot 1 = 6$$
$$\Delta^4 p(0) = 4! \cdot 1 = 24$$
$$\Delta^5 p(0) = 5! \cdot 0 = 0$$

Bevisszük a táblázatba, majd meghatározzuk a következő adagot, és így tovább:

x	**p(x)**	**Δ**	**Δ^2**	**Δ^3**	**Δ^4**	**Δ^5**
	4					
		−1				
			−16			
				6		
					24	
						0

x	p(x)	Δ	Δ^2	Δ^3	Δ^4	Δ^5
	4					
		−1				
	3		**−16**			
		−17		**6**		
			−10		**24**	
				30		**0**
					24	

x	p(x)	Δ	Δ^2	Δ^3	Δ^4	Δ^5
	4					
		−1				
	3		**−16**			
		−17		**6**		
	−14		−10		**24**	
		−27		30		**0**
			20		24	
				54		

A lépéseket addig ismételjük, amíg a táblázat kész nincs:

x	p(x)	Δ	Δ^2	Δ^3	Δ^4	Δ^5
	4					
		−1				
	3		**−16**			
		−17		**6**		
	−14		−10		**24**	
		−27		30		**0**
	−41		20		24	
		−7		54		
	−48		74			
		67				
	19					

Az anti-differencia pedig:

$$\Delta^{-1}p_n(x) = \frac{(x)^{(5)}}{5} + \frac{(x)^{(4)}}{4} - 8\frac{(x)^{(3)}}{3} - \frac{(x)^{(2)}}{2} + 4(x)^{(1)} + C$$

ahol C egy konstanst jelöl

Ezzel tulajdonképpen végre is hajtottuk az integrálást.

3.7 Példa alapján igen könnyen belátható, hogy egy $p_n(x)$ az $a \leq x \leq a + (n-1)$ intervallumon vett sorozatösszege:

$$\sum_{x=\alpha}^{\alpha+(n-1)h} p_n(x) = p_n(\alpha) + p_n(\alpha+h) + p_n(\alpha+2h) + \cdots + p_n[\alpha+(n-1)h] \tag{3.46.}$$

Kompaktabb formában pedig felírható így is:

$$\sum_{x=\alpha}^{\alpha+(n-1)h} p_n(x) = \Delta^{-1}p_n(x)|_{\alpha}^{\alpha+nh} \tag{3.47.}$$

Ez pedig tulajdonképpen nem jelent mást, mint hogy egy a már jól ismert $\int_a^b f(x)dx = F(b) - F(a)$ határozott integrál értéke egy szorzatösszeggel közelíthető. Ez különösen fontos lesz majd a későbbi fejezetekben.

3.3 Interpoláció

Eddig azt tárgyaltuk, hogy ha van egy számhalmazunk, akkor azokat hogyan tudjuk egy függvénnyel közelíteni, hogy az a lehető legkisebb hibát adja a pontok és a függvény értékei közt. Az approximációs eljárások során a közelítő függvénynek nem volt szükséges átmennie minden egyes ponton, hiszen a kiinduló számhalmaz vagy nagyon sok elemet tartalmaz, vagy pedig valamekkora szórást, így ha ténylegesen át kellene haladnunk minden egyes ponton, akkor nagy mértékben megnövelnénk a számítási igényt, ráadásul az eredmény sem feltétlenül egyezne meg az elvárásainkkal.

Viszont akad számos olyan helyzet, amikor a bemenő pontjainkon a keresett függvénynek át kell haladnia, ez esetben interpolációs eljárásokat alkalmazunk. Az interpoláció tulajdonképpen nem más, mint két számhalmaz elemei közötti kapcsolat sokaságának leírása; egy olyan matematikai közelítő módszer, amely egy függvény általunk nem ismert értékeit az ismert értékek alapján közelíti. Bármilyen interpolációs eljárást is veszünk, annak mindig teljesítenie kell három feltételt:

- át kell haladnia bizonyos pontokon
- további, speciális feltételeknek kell megfelelnie, pl.: meredekségi, görbületi feltételek
- a könnyű kezelhetőség miatt polinomokat alkalmazzunk.

Tehát ha a problémát matematikailag szeretnénk megfogalmazni, akkor azt mondhatjuk, hogy ha egy adott $f: \mathbb{R} \to \mathbb{R}$ függvény értékeit ismerjük:

$$y_i = f(x_i), \quad i = 1, 2, \dots, n \tag{3.48.}$$

az $a \leq x_1 \leq x_2 \leq \cdots \leq x_n \leq x_{n+1} \leq b$ pontokban ($n+1$ darab pont), akkor tulajdonképpen nem mást kell tennünk, mint keresni azt a $p: \mathbb{R} \to \mathbb{R}$ függvényt, mely kielégíti a

$$p(x_i) = y_i \quad i = 1, 2, \dots, n \tag{3.49.}$$

interpolációs feltételeket.

Ezen feladat során az $\{x_i\}_{i=1}^{n+1}$ pontokat interpolációs alappontoknak hívjuk. Hogy ha az interpolációs feltételeket jól határoztuk meg és azok teljesülnek, akkor várhatóan a $p(x)$ függvény jól fogja közelíteni $f(x)$ függvényt az alappontok által meghatározott intervallumokon.

Amennyiben az interpolációs függvényt arra használjuk, hogy segítségével (x_1, x_{n+1}) intervallumon kívüli közelítést hajtunk végre, akkor **extrapolációról** beszélünk.

A polinom interpoláció során az $n+1$ darab alappont közelítésére a következő alakú, n-ed fokú polinomot használjuk:

$$p(x) = a_{n+1}x^n + a_n x^{n-1} + \cdots + a_3 x^2 + a_2 x + a_1 \tag{3.50.}$$

mely az említett módon minden alapponton áthalad. A polinom meghatározására különböző módszerek léteznek, a két leggyakrabban használt a Lagrange és Newton interpolációs eljárások matematikai körökben. Mérnöki alkalmazásban pedig a Spline interpoláció terjedt el. A következőkben ezeket fogjuk átnézni.

3.3.1 Lagrange interpoláció

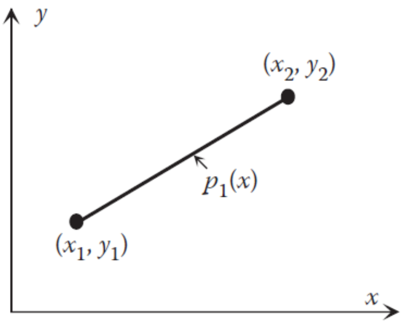
Ahhoz, hogy eljussunk a Lagrange interpoláció általános alakjához, először vegyük azt az esetet, amikor a Lagrange féleinterpoláló polinomnak át kell haladnia (x_1, y_1) és (x_2, y_2) pontokon, miközben a következő alakot ölti fel:

$$p_1(x) = L_1(x)y_1 + L_2(x)y_2 \tag{3.51.}$$

ahol $L_1(x)$ és $L_2(x)$ a Lagrange féle együtthatók, és a következő módon adhatók meg:

$$L_1(x) = \frac{x - x_2}{x_1 - x_2}, \quad L_2(x) = \frac{x - x_1}{x_2 - x_1} \tag{3.52.}$$

Ekkor, ha $L_1(x_1) = 1$ és $L_1(x_2) = 0$, miközben $L_2(x_1) = 0$ és $L_2(x_2) = 1$. Vagyis másképpen kifejezve $p_1(x_1) = y_1$ és $p_1(x_2) = y_2$, azaz a polinom esetünkben egy egyenes, mely pont áthalad a két ponton, ahogy a következő ábrán is látható.



6. *ábra: Elsőfokú Lagrange interpolációs polinom*

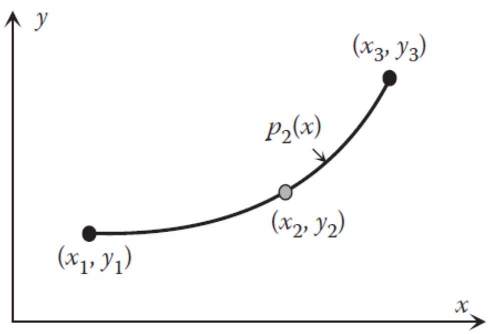
Ha a másodfokú esetet vesszük, akkor a Lagrange interpoláció a következőképpen írható fel (x_1, y_1), (x_2, y_2) és (x_3, y_3) pontokra:

$$p_2(x) = L_1(x)y_1 + L_2(x)y_2 + L_3(x)y_3 \tag{3.53.}$$

ahol a Lagrange együtthatók:

$$L_1(x) = \frac{(x - x_2)(x - x_3)}{(x_1 - x_2)(x_1 - x_3)}, \quad L_2(x) = \frac{(x - x_1)(x - x_3)}{(x_2 - x_1)\,(x_2 - x_3)}, \\ L_3(x) = \frac{(x - x_1)(x - x_2)}{(x_3 - x_1)\,(x_3 - x_2)} \tag{3.54.}$$

Ekkor $L_1(x_1) = 1 = L_2(x_2) = L_3(x_3)$, míg a többi együttható: $L_i(x_j) = 0$ minden $i \neq j$ értékre. Ez természetesen nem jelenthet mást, mint: $p_2(x_1) = y_1, p_2(x_2) = y_2$ és $p_2(x_3) = y_3$, vagyis a másodfokú polinom 3 ponton fog átmenni, ahogy a következő ábrán is megfigyelhető.



7. *ábra: Másodfokú Lagrange interpolációs polinom*

Ebből a két példából már remélhetőleg leszűrhető a minta, melyet a következőképpen foglalhatunk össze: az n-edfokú Lagrange interpoláló polinom, mely $n+1$ ponton megy át $(x_1, y_1), \ldots, (x_{n+1}, y_{n+1})$ az alábbi alakban írható le:

$$p_n(x) = L_1(x)y_1 + \cdots + L_{n+1}(x)y_{n+1} = \sum_{i=1}^{n+1} L_i(x)y_i \qquad (3.55.)$$

ahol a Lagrange együtthatók:

$$L_i(x) = \prod_{\substack{j=1 \\ j\neq i}}^{n+1} \frac{x - x_j}{x_i - x_j} \qquad (3.56.)$$

Nézzünk egy példát, hogyan használható fel ez egy feladat megoldására!

3.8. Példa

Feladat:

Másodfokú Lagrange módszer segítségével keressük meg a következő pontok interpolációs polinomját, majd interpoláljuk ki $x = 0{,}3$ helyen a várható értéket!

i	x_i	y_i
1	0,1	0,12
2	0,5	0,47
3	0,9	0,65

Megoldás:

A három Lagrange együttható a következőképpen számítható:

$$L_1(x) = \frac{(x - x_2)(x - x_3)}{(x_1 - x_2)(x_1 - x_3)} = \frac{(x - 0{,}5)(x - 0{,}9)}{(0{,}1 - 0{,}5)(0{,}1 - 0{,}9)}$$

$$L_2(x) = \frac{(x - x_1)(x - x_3)}{(x_2 - x_1)(x_2 - x_3)} = \frac{(x - 0{,}1)(x - 0{,}9)}{(0{,}5 - 0{,}1)(0{,}5 - 0{,}9)}$$

$$L_3(x) = \frac{(x - x_1)(x - x_2)}{(x_3 - x_1)(x_3 - x_2)} = \frac{(x - 0{,}1)(x - 0{,}5)}{(0{,}9 - 0{,}1)(0{,}9 - 0{,}5)}$$

Az együtthatók ismeretében a másodfokú polinom a következőképp írható fel:

$$p_2(x) = L_1(x)y_1 + L_2(x)y_2 + L_3(x)y_3 =$$

$$= \frac{(x-0,5)(x-0,9)}{(0,1-0,5)(0,1-0,9)}(0,12) + \frac{(x-0,1)(x-0,9)}{(0,5-0,1)(0,5-0,9)}(0,47) + \frac{(x-0,1)(x-0,5)}{(0,9-0,1)(0,9-0,5)}(0,65) =$$

$$= -0,5312x^2 + 1,1937x + 0,0059$$

Vagyis visszahelyettesítve $x = 0,3$ helyen: $p_2(0,3) = 0,3162$.

Kiterjesztve magasabb rendű problémára, vegyünk egy példát Excel-ben megoldva.

3.9. Példa

Feladat:

Lagrange módszer segítségével keressük meg a következő pontok interpolációs polinomját, majd interpoláljuk ki $x = 2$ helyen a várható értéket!

i	x_i	y_i
1	-1	3
2	0	-2
3	0,5	-0,375
4	1	3
5	2,5	16,125
6	3	19

Megoldás:

Hozzunk létre egy „célszerű” táblázatot, ahol csak a B6:C11 cellákat kell megadni, a többi cella értéke képlettel határozható meg, ahogy a beírt képletek is mutatják:

	A	B	C	D	E	F	G	H	I	J	K	L
1	Lagrange interpolácia											
2												
3	Interpoláljunk a következő helyen			2		Kezdőérték						
4										Számláló	Nevező	Hányados
5		x	f(x)	$x-x_1$	$x-x_2$	$x-x_3$	$x-x_4$	$x-x_5$	$f(x_0)$			
6	x_0	-1,000	3,000	2,000	1,500	1,000	-0,500	-1,000	3,000	4,500		
7	x_1	0,000	-2,000	x_0-x_1	x_0-x_2	x_0-x_3	x_0-x_4	x_0-x_5				-0,107
8	x_2	0,500	-0,375	-1,000	-1,500	-2,000	-3,500	-4,000			-42,000	
9	x_3	1,000	3,000	$x-x_0$	$x-x_2$	$x-x_3$	$x-x_4$	$x-x_5$	$f(x_1)$			
10	x_4	2,500	16,125	3,000	1,500	1,000	-0,500	-1,000	-2,000	-4,500		
11	x_5	3,000	19,000	x_1-x_0	x_1-x_2	x_1-x_3	x_1-x_4	x_1-x_5				-1,200
12				1,000	-0,500	-1,000	-2,500	-3,000			3,750	
13				$x-x_0$	$x-x_1$	$x-x_3$	$x-x_4$	$x-x_5$	$f(x_2)$			
14				3,000	2,000	1,000	-0,500	-1,000	-0,375	-1,125		
15				x_2-x_0	x_2-x_1	x_2-x_3	x_2-x_4	x_2-x_5				0,600
16				1,500	0,500	-0,500	-2,000	-2,500			-1,875	
17				$x-x_0$	$x-x_1$	$x-x_2$	$x-x_4$	$x-x_5$	$f(x_3)$			
18				3,000	2,000	1,500	-0,500	-1,000	3,000	13,500		
19				x_3-x_0	x_3-x_1	x_3-x_2	x_3-x_4	x_3-x_5				4,500
20				2,000	1,000	0,500	-1,500	-2,000			3,000	
21				$x-x_0$	$x-x_1$	$x-x_2$	$x-x_3$	$x-x_5$	$f(x_4)$			
22				3,000	2,000	1,500	1,000	-1,000	16,125	-145,125		
23				x_4-x_0	x_4-x_1	x_4-x_2	x_4-x_3	x_4-x_5				11,057
24				3,500	2,500	2,000	1,500	-0,500			-13,125	
25				$x-x_0$	$x-x_1$	$x-x_2$	$x-x_3$	$x-x_4$	$f(x_5)$			
26				3,000	2,000	1,500	1,000	-0,500	19,000	-85,500		
27				x_5-x_0	x_5-x_1	x_5-x_2	x_5-x_3	x_5-x_4				-2,850
28				4,000	3,000	2,500	2,000	0,500			30,000	
29												
30										f(2)=	Sum=	12,000

A Lagrange interpolációt ugyanakkor muszáj a helyén kezelnünk, mivel megvannak a hátrányai. Ha a Lagrange polinom fokszámát elkezdjük növelni, akkor az alacsonyabb fokszámú polinomokról semmilyen információ nem adódik át az új polinom megszerkesztésekor. Ez különösen kellemetlen lehet két esetben:

- ha a keresett polinom fokszáma nem ismert,
- ha a kiindulási pontokhoz újabbakat teszünk hozzá.

Ezen esetekben sokkal pontosabb megoldást ad a Newton interpoláció.

3.3.2 Newton-féle interpolációs polinom

A Newton interpolációs polinomok rekurzív módon a következőképpen írhatók fel:

$$
\begin{aligned}
p_1(x) &= a_1 + a_2(x - x_1) \\
p_2(x) &= a_1 + a_2(x - x_1) + a_3(x - x_1)(x - x_2) \\
&= p_1(x) + a_3(x - x_1)(x - x_2) \\
p_3(x) &= p_2(x) + a_4(x - x_1)(x - x_2)(x - x_3) \\
&\dots \\
p_n(x) &= P_{n-1}(x) + a_{n+1}(x - x_1)(x - x_2)\dots(x - x_n)
\end{aligned}
\tag{3.57.}
$$

ahol a $a_1, a_2, \dots, a_{n+1}$ együtthatók meghatározása a következő módon történik: Mivel $p_1(x)$-nek egyeznie kell a megadott pontokkal ((x_1, y_1) és (x_2, y_2)), így:

$$p_1(x_1) = y_1,\ \ p_1(x_2) = y_2 \tag{3.58.}$$

vagyis

$$\begin{matrix} a_1 + a_2(x_1 - x_1) = y_1 \\ a_1 + a_2(x_2 - x_1) = y_2 \end{matrix} => a_1 = y_1,\ \ a_2 = \frac{y_2 - y_1}{x_2 - x_1} \tag{3.59.}$$

Hasonlóan $p_2(x)$-nek meg kell egyeznie a megadott pontokkal ((x_1, y_1), (x_2, y_2) és (x_3, y_3)), vagyis:

$$p_2(x_1) = y_1,\ \ p_2(x_2) = y_2, p_2(x_3) = y_3 \tag{3.60.}$$

a_1 és a_2 együtthatókat továbbra is úgy lehet meghatározni, mint előbb [(3.59.) egyenlet], a_3-at pedig:

$$a_3 = \frac{\frac{y_3 - y_2}{x_3 - x_2} - \frac{y_2 - y_1}{x_2 - x_1}}{x_3 - x_1} \tag{3.61.}$$

Ezt az eljárást folytatva megkaphatjuk az összes hiányzó együtthatót. Ezen együtthatók meghatározásának a legkézenfekvőbb módszere az Newton-féle osztott differencia alkalmazása. Két pontra számítva az első osztott differencia az őket összekötő egyenest adja:

$$f[x_2, x_1] = \frac{y_2 - y_1}{x_2 - x_1} = a_2 \tag{3.62.}$$

Ha három pontra írjuk fel, akkor:

$$f[x_3, x_2, x_1] = \frac{f[x_3, x_2] - f[x_2, x_1]}{x_3 - x_1} = \frac{\frac{y_3 - y_2}{x_3 - x_2} - \frac{y_2 - y_1}{x_2 - x_1}}{x_3 - x_1} = a_3 \tag{3.63.}$$

A módszer általánosításával a következő képlettel írhatjuk fel a k-adik osztott differenciát:

$$f[x_{k+1}, x_k, \dots, x_2, x_1] = \frac{f[x_{k+1}, x_k, \dots, x_2] - f[x_k, \dots, x_2, x_1]}{x_{k+1} - x_1} = a_{k+1} \qquad (3.64.)$$

Vagyis az n-ed fokú Newton-féle interpolációs polinom $(x_1, y_1), \dots, (x_{n+1}, y_{n+1})$ pontokra az alábbi módon írható fel:

$$\begin{aligned} p_n(x) &= a_1 + a_2(x - x_1) + a_3(x - x_1)(x - x_2) + \cdots + \\ &+ a_{n+1}(x - x_1)(x - x_2) \dots (x - x_n) \end{aligned} \qquad (3.65.)$$

ahol $a_1, \dots, a_{n+1}$ együtthatók sorban $y_1, f[x_2, x_1], \dots, f[x_{n+1}, x_1]$, és számításukra az osztott differencia táblázat a legjobb megoldás.

3.10. Példa

Feladat:

Keressük meg a negyedrendű Newton-féle interpolációs polinomját az alábbi adatoknak, majd nézzük meg, hogy $x = 0{,}7$-nél milyen értékre számíthatunk!

i	x_i	y_i
1	0	0
2	0,1	0,1210
3	0,2	0,2258
4	0,5	0,4650
5	0,8	0,6249

Megoldás:

Az osztott differencia táblázat megalkotása az első feladatunk, ez a később mutatott módon néz ki.

A (3.64.) egyenlet alapján pedig a negyedrendű Newton-féle interpoláló polinom a következő alakot ölti:

$$p_4(x) = a_1 + a_2(x - x_1) + a_3(x - x_1)(x - x_2) + a_4(x - x_1)(x - x_2)(x - x_3) + a_5(x - x_1)(x - x_2)(x - x_3)(x - x_4) =$$

$$= 0 + 1{,}21(x - 0) - 0{,}81(x - 0)(x - 0{,}1) + 0{,}3664(x - 0)(x - 0{,}1)(x - 0{,}2) - $$
$$-0{,}1254(x - 0)(x - 0{,}1)(x - 0{,}2)(x - 0{,}5) =$$
$$= -0{,}1254x^4 + 0{,}4667x^3 - 0{,}9412x^2 + 1{,}2996x$$

Ezt a polinomot használva pedig már könnyedén számolni tudjuk a keresett értéket:

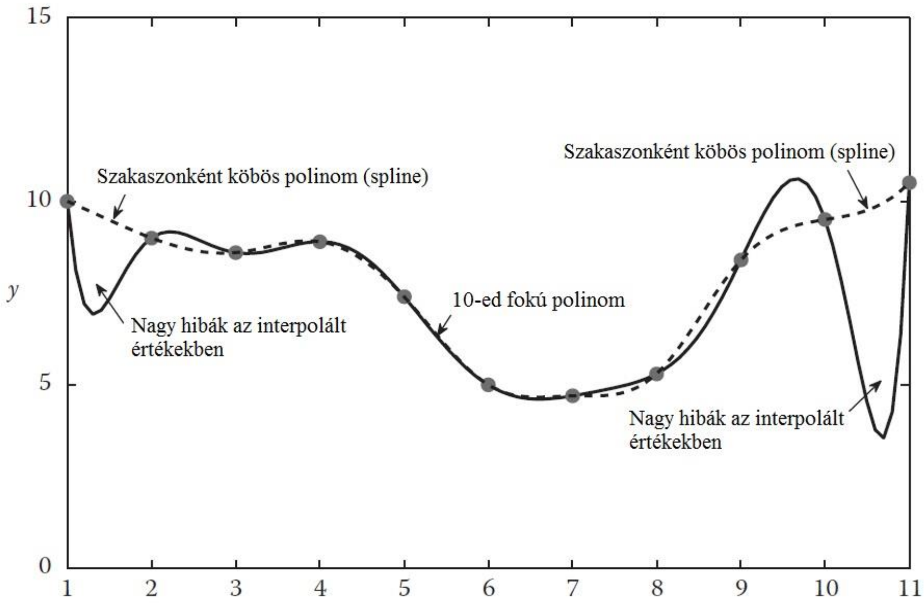
$$p_4(0{,}7) = 0{,}5784$$

A differencia táblázat pedig:

x_i	y_i	Első osztott differencia	Második osztott differencia	Harmadik osztott differencia	Negyedik osztott differencia
0	$\overset{a_1}{\boxed{0}}$				
		$\frac{0.1210-0}{0.1-0} = \overset{a_2}{\boxed{1.2100}}$			
0.1	0.1210		$\frac{1.0480-1.2100}{0.2-0} = \overset{a_3}{\boxed{-0.8100}}$		
		$\frac{0.2258-0.1210}{0.2-0.1} = 1.0480$		$\frac{-0.6268-(-0.8100)}{0.5-0} = \overset{a_4}{\boxed{0.3664}}$	
0.2	0.2258		$\frac{0.7973-1.0480}{0.5-0.1} = -0.6268$		$\frac{0.2661-0.3664}{0.8-0} = \overset{a_5}{\boxed{-0.1254}}$
		$\frac{0.4650-0.2258}{0.5-0.2} = 0.7973$		$\frac{-0.4405-(-0.6268)}{0.8-0.1} = 0.2661$	
0.5	0.4650		$\frac{0.5330-0.7973}{0.8-0.2} = -0.4405$		
		$\frac{0.6249-0.4650}{0.8-0.5} = 0.5330$			
0.8	0.6249				

3.3.3 Spline interpoláció

Eddig ahhoz, hogy $n+1$ darab pontra egy polinomot illeszthessünk, egy n-ed fokú polinomot használtunk. Például ha van 11 bemenő pontunk, akkor a Lagrange interpoláció során egy 10-ed fokú polinomot fogunk kapni. Ez a megoldás jól működik is, amennyiben pontjaink száma alacsony, de ha már 101 vagy 1001 pontot kellene közelítenünk, érezhető, hogy van egyszerűbb megoldás is a 100-ad vagy 1000-ed fokú polinomoknál. Ráadásul ezekben az esetekben már a generált hiba is jóval nagyobb lesz. Ha például 11 pont közelítésére egy 10-ed fokú polinomot veszünk, a magasabb rendű polinom csúcsai miatt jelentős eltéréseket képes produkálni, ahogy ezt a 8. ábra is szemlélteti. Természetesen minél magasabb az interpoláló polinom fokszáma, az abszolút hiba értéke is úgy növekszik.



8. ábra: 11 pont közelítése 10-ed fokú polinommal és szakaszonként köbös polinomokkal (spline-okkal)

Ezen jelenség kiküszöböléseként született meg az a megoldás, miszerint nagyobb számú pontot érdemes nem egy magas fokszámú polinommal, hanem szakaszonként több kisebb fokszámú polinommal közelíteni. Ezek az alacsony fokszámú polinomok az úgynevezett spline-ok. A spline elnevezés egy vékony, hajlékony vonalzóra utal, melyet valamilyen rugalmas fából készítettek, később acélból. A vonalzót két végén rögzítették, közepén pedig súlyokkal terhelték, ezzel különböző görbületű görbéket hoztak létre, melyeket elsősorban hajó- és csónaképítések során használtak a műszaki rajzolók. Ekkor még matematikailag nem tudták leírni ezeket a függvényeket, de izgalomra semmi ok, azóta felülkerekedett az emberiség ezen elmaradottságán!

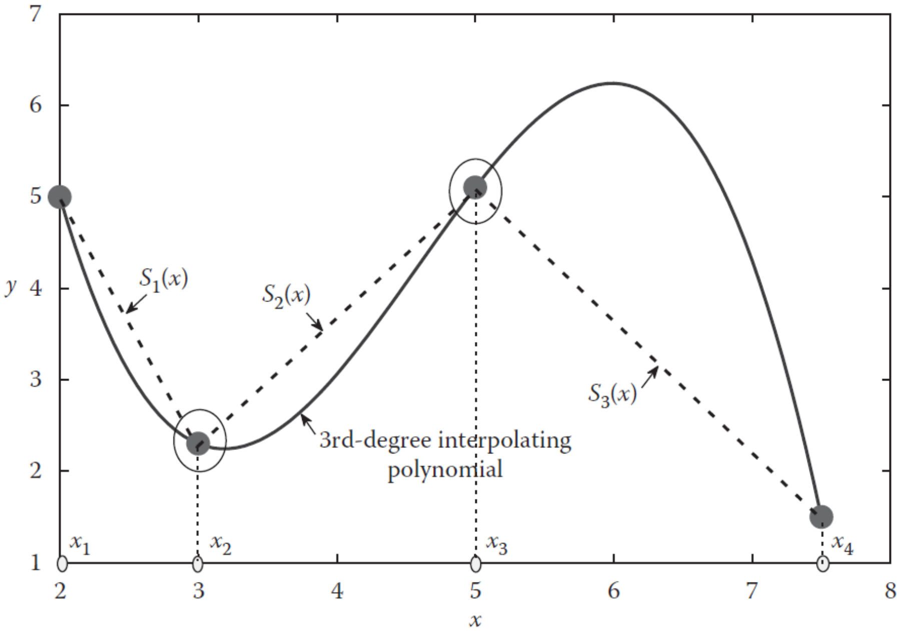
A leggyakrabban alkalmazott spline a köbös spline, mely igen pontos leírást és átmenetet képes biztosítani két intervallum között, ahogy a 8. ábra is reprezentálja. Ugyanezen az ábrán látható – különösebb magyarázat nélkül – hogy miért is szeretjük alkalmazni őket, különösen az egy, magasfokú interpoláló polinommal szemben.

3.3.3.1 Lineáris spline-ok

A lineáris spline-ok jelentik a legegyszerűbb megközelítést, mivel ebben az esetben elsőrendű polinommal – lineáris függvényekkel kötjük össze a pontjainkat. Amennyiben rendelkezünk 4 darab ponttal (legyenek (x_1, y_1), (x_2, y_2), (x_3, y_3) és (x_4, y_4)), ha visszanyúlunk a Lagrange módszerig, akkor a 3 lineáris spline a következő módon írható fel:

$$
\begin{aligned}
S_1(x) &= \frac{x - x_2}{x_1 - x_2} y_1 + \frac{x - x_1}{x_2 - x_1} y_2, \quad x_1 \le x \le x_2 \\
S_2(x) &= \frac{x - x_3}{x_2 - x_3} y_2 + \frac{x - x_2}{x_3 - x_2} y_3, \quad x_2 \le x \le x_3 \\
S_3(x) &= \frac{x - x_4}{x_3 - x_4} y_3 + \frac{x - x_3}{x_4 - x_3} y_4, \quad x_3 \le x \le x_4
\end{aligned}
\tag{3.66.}
$$

Ha vetünk egy pillantást a lineáris splinokra (lásd 9. ábra), teljes joggal merülhet fel a gyanú, hogy ezzel már találkoztunk valahol, és tulajdonképpen nem is tévednénk, mivel ez nem más, mint a 3.3.1 fejezet más köntösbe bújtatva. Ebből következően pedig adódnak is a hátrányai, hiszen ez a lineáris spline nem képes a szükséges simasággal lekövetni a bemeneti pontokat, ráadásul a pontokban magukban mindig törés jön létre a függvény képében. Ez abból adódik, hogy a lineáris függvények első deriváltja a pontokban nem azonos (ha ezt kikötnénk, nem tudnánk lineárisan közelíteni a pontokat). Pont ezért fogjuk a későbbiekben azt mondani, hogy az első deriváltaknak márpedig egyeznie kell, hiszen komoly ember ilyen interpolációt mégsem hajthat végre. Ezt viszont csak úgy tudjuk megtenni, ha magasabb rendű polinomokkal dolgozunk, hogy azok első deriváltja megegyezhessen minden egyes bemeneti pontban. Ez például a másodfokú polinomok esetében megoldható, de már most előre vetíthető, hogy a következő problémánk majd a második deriváltak egyezősége lesz.



9. ábra: 4 pont közelítése lineáris spline-okkal

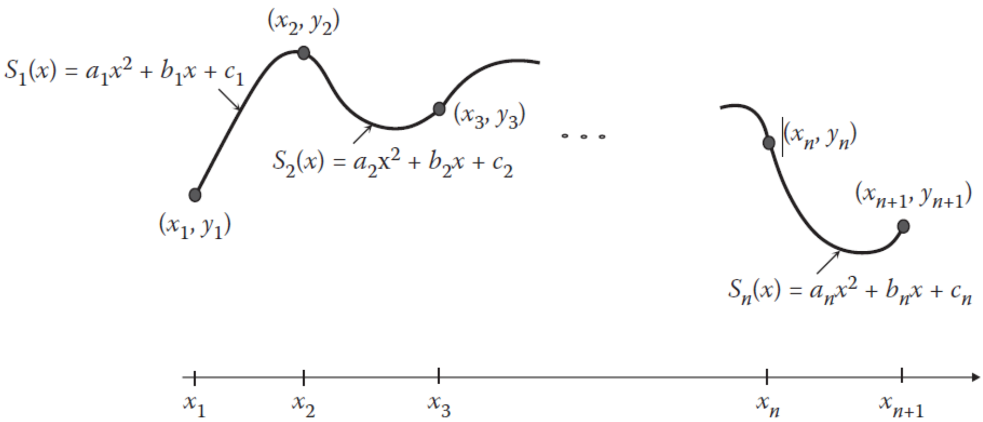
3.3.3.2 Négyzetes spline-ok

A négyzetes spline-ok tulajdonképpen parabolikus függvényeket (spline-okat) használnak a bemeneti pontok közelítésére minden egyes intervallumon. Feltételezve, hogy $n+1$ darab bemeneti pontunk van, melyek $(x_1, y_1), \dots,$ (x_{n+1}, y_{n+1}), ezek n intervallumra oszthatók, vagyis n darab négyzetes polinomot kell felírnunk, lásd 10. ábra. A négyzetes polinomok a következő formában írhatók fel:

$$S_i(x) = a_i x^2 + b_i x + c_i, \quad i = 1,2, \dots, n \tag{3.67.}$$

ahol a_i, b_i, c_i $(i = 1,2, \dots, n)$ ismeretlen együtthatók, melyeket meg kell határoznunk. Mivel n darab polinom van, mindegyik 3 darab együtthatóval, így összesen $3n$ ismeretlen együttható várja, hogy megfejtsük őket. Ahhoz, hogy ezt megtehessük, pontosan $3n$ darab egyenletet kell meghatároznunk.

$$\left.\begin{aligned} S_1(x_1) &= y_1 \\ S_n(x_{n+1}) &= y_{n+1} \end{aligned}\right\} \text{2 egyenlet}$$



10. ábra: Négyzetes spline-ok

A $3n$ darab egyenlet meghatározásához a következőképp jutunk el, vagyis hogy a következő feltételrendszert állítjuk fel:

Függvények értékei a végpontokban

Az első polinomnak, $S_1(x)$-nek át kell mennie (x_1, y_1) ponton, az utolsónak – $S_n(x)$ pedig (x_{n+1}, y_{n+1}) ponton:

$$\begin{gathered} S_1(x) = y_1 = a_1x^2 + b_1x_1 + c_1 \\ S_n(x_{n+1}) = y_{n+1} = a_nx_{n+1}^2 + b_nx_{n+1} + c_n \end{gathered} \qquad (3.68.)$$

Függvény értékei közbenső pontokban

Közbenső pontokban 2 feltételt szeretnénk teljesíteni: a polinomoknak át kell mennie a bemenő pontokon, illetve a szomszédos polinomoknak egyeznie kell az alábbi pontokban, ezzel $n-1$ darab egyenletet generálva:

$$\begin{gathered} S_1(x_2) = y_2,\ \ S_2(x_3) = y_3, \dots, S_{n-1}(x_n) = y_n \\ S_2(x_2) = y_2,\ \ S_3(x_3) = y_3, \dots, S_n(x_n) = y_n \end{gathered} \qquad (3.69.)$$

Vagyis a következőképpen írhatjuk fel általánosan:

$$\begin{gathered} S_i(x_{i+1}) = y_{i+1} = a_ix_{i+1}^2 + b_ix_{i+1} + c_i,\ \ i = 1,2, \dots, n-1 \\ S_i(x_i) = y_i = a_ix_i^2 + b_ix_i + c_i,\ \ i = 2,3, \dots, n-1 \end{gathered} \qquad (3.70.)$$

Első deriváltak értéke közbenső pontokban

A közbenső pontokban vett első deriváltak értékeinek szintén egyeznie kell, melyet matematikailag a következő módon tudunk kifejezni $n-1$ darab egyenlettel:

$$S'_1(x_2) = S'_2(x_2),\;\; S'_2(x_3) = S'_3(x_3), \dots, S'_{n-1}(x_n) = S'_n(x_n) \qquad (3.71.)$$

Általánosan kifejezve:

$$S'_i(x_{i+1}) = S'_{i+1}(x_{i+1}),\;\; i = 1,2, \dots, n-1 \qquad (3.72.)$$

Mivel $S'_i(x) = 2a_i x + b_i$, az előző egyenlet másképp felírva:

$$2a_i x_{i+1} + b_i = 2a_{i+1} x_{i+1} + b_{i+1},\;\; i = 1,2, \dots, n-1 \qquad (3.73.)$$

Ha ezeket az egyenleteket összeadjuk, akkor láthatjuk, hogy $2 + (n-1) + (n-1) + (n-1) = 3n - 1$ darab egyenletet hoztunk össze, ami már igen szép szám, de még mindig szükségünk van egy további egyenletre, hogy megoldhatóvá váljon a rendszerünk.

Második derivált értéke a bal oldali végpontban nulla

Ez azt jelenti, hogy:

$$S''_1(x_1) = 0 = 2a_1 \qquad (3.74.)$$

Ezzel végül sikerült összegereblyézni a szükséges $3n$ darab egyenletet. Ha kissé leegyszerűsítve nézzük, akkor tulajdonképpen az történt, hogy egy egyenlettel kimondtuk, $a_1 = 0$, míg a maradék $3n-1$ darab ismeretlent egy $3n-1$ elemű egyenletrendszerrel kerestük meg.

3.11. Példa

Határozzuk meg a négyzetes spline-okat a következő bemenő pontokra:

x_i	y_i
2	5
3	2,3
5	5,1
7,5	1,5

Megoldás:

Mivel 4 darab pont van, így $n = 3$-ra adódik, vagyis 3 parabolikus spline-t akarunk megkeresni összesen 9 ismeretlen együtthatóval. Elsőként tudjuk, hogy $a_1 = 0$ értékre fog adódni. A maradék 8 egyenlet pedig a (3.68.)-(3.73.) egyenletek megoldása lesz. (3.68.) egyenlet alapján:

$$a_1(2)^2 + b_1(2) + c_1 = 5$$
$$a_3(7{,}5)^2 + b_3(7{,}5) + c_3 = 1{,}5$$

A (3.70.) egyenlet alapján:

$$a_1(3)^2 + b_1(3) + c_1 = 2{,}3$$
$$a_2(5)^2 + b_2(5) + c_2 = 5{,}1$$
$$a_2(3)^2 + b_2(3) + c_2 = 2{,}3$$
$$a_3(5)^2 + b_3(5) + c_3 = 5{,}1$$

Végül pedig (3.73.) egyenlet alapján:

$$2a_1(3) + b_1 = 2a_2(3) + b_2$$
$$2a_2(5) + b_2 = 2a_3(5) + b_3$$

Az egyszerűség kedvéért mátrixos formában felírva az egyenletrendszert:

$$\begin{bmatrix} 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 56{,}25 & 7{,}5 & 1 \\ 3 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 25 & 5 & 1 & 0 & 0 & 0 \\ 0 & 0 & 9 & 3 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 25 & 5 & 1 \\ 1 & 0 & -6 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 10 & 1 & 0 & -10 & -1 & 0 \end{bmatrix} \begin{bmatrix} b_1 \\ c_1 \\ a_2 \\ b_2 \\ c_2 \\ a_3 \\ b_3 \\ c_3 \end{bmatrix} = \begin{bmatrix} 5 \\ 1{,}5 \\ 2{,}3 \\ 5{,}1 \\ 2{,}3 \\ 5{,}1 \\ 0 \\ 0 \end{bmatrix}$$

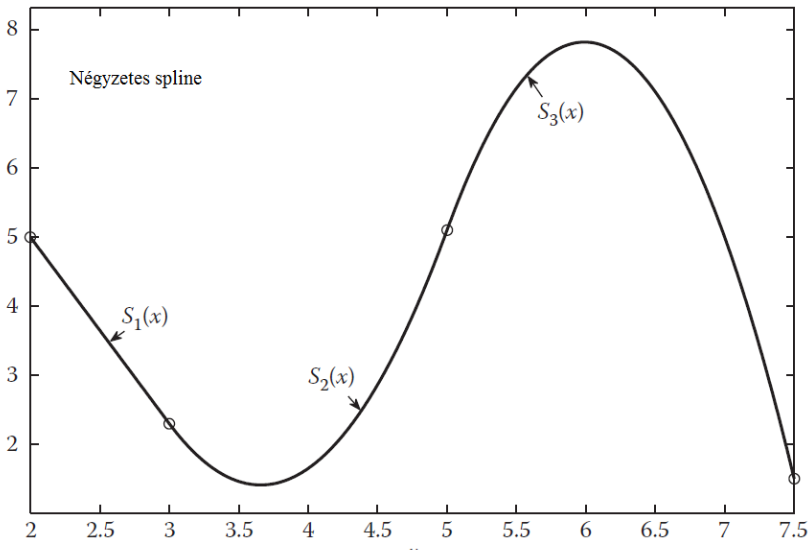
Megoldva az egyenletrendszert a következő értékeket kapjuk az együtthatókra:

$$a_1 = 0; \quad a_2 = 2{,}05; \quad a_3 = -2{,}776;$$
$$b_1 = -2{,}7; \quad b_2 = -15; \quad b_3 = 33{,}26;$$
$$c_1 = 10{,}4; \quad c_2 = 28{,}85; \quad c_3 = -91{,}8$$

Végezetül pedig a parabolikus spline-ok egyenleteit a következő módon adhatjuk meg:

$$S_1(x) = -2{,}7x + 10{,}4 \qquad 2 \le x \le 3$$
$$S_2(x) = 2{,}05x^2 - 15x + 28{,}85 \qquad 3 \le x \le 5$$
$$S_3(x) = -2{,}776x^2 + 33{,}26x - 91{,}8 \qquad 5 \le x \le 7{,}5$$

A feladat megoldása pedig ábrázolva a következő alakot fogja ölteni:



3.3.3.3 Köbös spline-ok

Köbös spline-ok esetén harmadfokú polinomok felhasználásával interpoláljuk ki az egyes intervallumok közötti részeket. Feltételezve $n+1$ darab bemeneti pontot, melyek $(x_1, y_1), \dots, (x_{n+1}, y_{n+1})$, kapunk n darab intervallumot, vagyis n darab köbös polinomot. Mindegyik polinom a következő általános alakban írható fel:

$$S_i(x) = a_i(x - x_i)^3 + b_i(x - x_i)^2 + c_i(x - x_i) + d_i, \quad i = 1,2, \dots, n \qquad (3.75.)$$

ahol a_i, b_i, c_i, d_i $(i = 1,2, \dots, n)$ az ismeretlen, ezáltal meghatározandó együtthatók. Mivel n darab polinom van, mindegyik 4 együtthatóval, így összesen $4n$ darab ismeretlenre kell rájönni, hogy mit is takar, vagyis $4n$ egyenletet kell összetákolni most. Ehhez hasonlóan az előző esethez szintén kényszereket kell definiálnunk.

A spline átmegy a kezdő-, vég- és közbenső pontokon, továbbá a közbenső pontokban a szomszédos spline-ok értéke megegyezik:

$$\begin{gathered} S_1(x_1) = y_1, \quad S_n(x_{n+1}) = y_{n+1} \\ S_{i+1}(x_{i+1}) = S_i(x_{i+1}), \quad i = 1,2, \dots, n-1 \\ S_i(x_i) = y_i, \quad i = 2,3, \dots, n \end{gathered} \qquad (3.76.)$$

Az első deriváltak értékei megegyeznek a közbenső pontokban ($n-1$ egyenlet):

$$S_i'(x_{i+1}) = S_{i+1}'(x_{i+1}), \;\; i = 1,2, \dots, n-1 \qquad (3.77.)$$

A második deriváltak értékei szintén megegyeznek a közbenső pontokban ($n-1$ egyenlet):

$$S_i''(x_{i+1}) = S_{i+1}''(x_{i+1}), \;\; i = 1,2, \dots, n-1 \qquad (3.78.)$$

Ezekkel összesen $4n-2$ egyenletet szedtünk össze, tehát még 2 híján vagyunk. Ezt a kettőt pedig peremfeltételek megadásával tudjuk definiálni. Ezen peremfeltételekkel tudjuk definiálni, hogy a spline milyen módon indul ki a kezdőpontból és hogyan érkezik meg a végpontba. Alapvetően két fajta peremfeltételt különböztetünk meg egymástól.

Rögzített peremfeltétel:

Az első és utolsó spline-ok (S_1 és S_n) indulása (x_1, y_1)-ből és érkezése (x_{n+1}, y_{n+1}) rögzítésre kerül:

$$S_1'(x_1) = p, \quad S_n'(x_{n+1}) = q \qquad (3.79.)$$

Szabad peremfeltétel:

Ez egy kicsit becsapós, hiszen itt a görbületre fogunk kényszert megadni:

$$S_1''(x_1) = 0, \quad S_n''(x_{n+1}) = 0 \qquad (3.80.)$$

Természetesen a rögzített peremfeltétel alkalmazása pontosabb közelítést fog adni, hiszen több információt tartalmaz a spline-ról, mint a szabad peremfeltétel. Ehhez persze többet is kell tudnunk magáról a spline-ról, hiszen értékét saját kezünkkel kell definiálni.

3.3.3.4 Köbös spline-ok létrehozása rögzített peremfeltétellel

A (3.75.)-(3.77.) egyenletek kiegészítve az (3.79.) egyenlettel meghatározzák a_i, b_i, c_i, d_i $(i = 1,2, \dots, n)$ együtthatókat. Sorjában haladva, először meg tudjuk határozni d_i együtthatókat, hiszen (3.75.) kifejezések első és utolsó egyenlete alapján látható, hogy $S_i(x_i) = d_i$ $(i = 1,2, \dots, n)$, továbbá (3.76.) egyenlet szerint $S_i(x_i) = y_i$ $(i = 1,2, \dots, n)$, így pedig:

$$d_i = y_i, \quad i = 1,2, \dots, n \qquad (3.81.)$$

Határozzuk meg az egyes pontok közötti távolságokat: $h_i = x_{i+1} - x_i$ $(i = 1,2, \dots, n)$. Ezt visszahelyettesítve (3.76.) egyenletbe (nem megfeledkezve róla, hogy $S_{i+1}(x_{i+1}) = d_{i+1}$) kifejezhető:

$$d_{i+1} = a_i h_i^3 + b_i h_i^2 + c_i h_i + d_i, \quad i = 1,2,\ldots,n-1 \tag{3.82.}$$

Ha definiáljuk, hogy $d_{n+1} = y_{n+1}$, akkor a fenti egyenlet a teljes $i = 1,2,\ldots,n$ tartományon érvényes lesz, hiszen $S_n(x_{n+1}) = y_{n+1}$. Eképp:

$$d_{i+1} = a_i h_i^3 + b_i h_i^2 + c_i h_i + d_i, \quad i = 1,2,\ldots,n \tag{3.83.}$$

Most pedig kapjuk elő $S_i(x)$ első deriváltját a (3.77.) egyenlet alapján:

$$c_{i+1} = 3a_i h_i^2 + 2b_i h_i + c_i, \quad i = 1,2,\ldots,n-1 \tag{3.84.}$$

Ha definiáljuk $c_{n+1} = S_n'(x_{n+1})$ értékét, akkor az előző egyenlet a teljes $i = 1,2,\ldots,n$ intervallumon érvényes lesz, vagyis:

$$c_{i+1} = 3a_i h_i^2 + 2b_i h_i + c_i, \quad i = 1,2,\ldots,n \tag{3.85.}$$

Már csak $S_i(x)$ második deriváltja árválkodik, ne is hagyjuk magára:

$$2b_{i+1} = 6a_i h_i + 2b_i, \quad i = 1,2,\ldots,n-1 \tag{3.86.}$$

Ha definiáljuk $b_{n+1} = \frac{1}{2} S_n''(x_{n+1})$ értékét, akkor az előző egyenlet a teljes $i = 1,2,\ldots,n$ intervallumon érvényes lesz, vagyis:

$$b_{i+1} = 3a_i h_i + b_i, \quad i = 1,2,\ldots,n \tag{3.87.}$$

A cél az lenne, hogy az egész egyenletrendszert b_i $(i = 1,2,\ldots,n+1)$ értékekre fejezzük ki. Ehhez meg kell oldani az előző (3.87.) egyenletet úgy, hogy behelyettesítjük $a_i = (b_{i+1} - b_i)/3h_i$ kifejezést (3.83.) és (3.85.) egyenletekbe $i = 1,2,\ldots,n$ értékekre, ezzel:

$$d_{i+1} = \frac{1}{3}(2b_i + b_{i+1})h_i^2 + c_i h_i + d_i, \quad i = 1,2,\ldots,n \tag{3.88.}$$

illetve

$$c_{i+1} = (b_i + b_{i+1})h_i + c_i, \quad i = 1,2,\ldots,n \tag{3.89.}$$

Kifejezzük (3.88.) egyenletből c_i-t:

$$c_i = \frac{d_{i+1} - d_i}{h_i} - \frac{1}{3}(2b_i + b_{i+1})h_i \tag{3.90.}$$

Átírjuk i-t $i-1$-re:

$$c_{i-1} = \frac{d_i - d_{i-1}}{h_{i-1}} - \frac{1}{3}(2b_{i-1} + b_i)h_{i-1} \tag{3.91.}$$

Továbbá ugyanezt csináljuk meg (3.89.) egyenlettel is:

$$c_i = (b_{i-1} + b_i)h_{i-1} + c_{i-1} \qquad (3.92.)$$

Kitartás, most már nincs sok hátra! Visszahelyettesítjük (3.90.) és (3.91.) egyenleteket (3.92.) egyenletbe, így $i = 2,3, \dots, n$-re:

$$b_{i-1}h_{i-1} + 2b_i(h_i + h_{i-1}) + b_{i+1}h_i = \frac{3(d_{i+1} - d_i)}{h_i} - \frac{3(d_i - d_{i-1})}{h_{i-1}} \qquad (3.93.)$$

Ezzel le is tudjuk írni azt a rendszert, melynek csak b_i $(i = 1,2, \dots, n+1)$ az ismeretlen együtthatói, hiszen d_i $(i = 1,2, \dots, n+1)$ értékei egyszerűen a bemenő pontok értékei, h_i $(i = 1,2, \dots, n)$ pedig meghatározza az osztásközt. Viszont nem mehetünk el szó nélkül amellett, hogy a (3.93.) egyenlet csak $n-1$ egyenletet generál $n+1$ ismeretlenre, vagyis további két egyenlet még mindig kell. Ezek fognak származni a rögzített peremfeltételekből.

Először is oldjuk meg (3.90.) egyenletet $i = 1$-re:

$$c_1 = \frac{d_2 - d_1}{h_1} - \frac{1}{3}(2b_1 + b_2)h_1 \qquad (3.94.)$$

Mivel korábban azt mondtuk, hogy a rögzített peremfeltétel miatt $c_1 = S_1'(x_1) = p$, így átírhatjuk előző egyenletünket:

$$(2b_1 + b_2)h_1 = \frac{3(d_2 - d_1)}{h_1} - 3p \qquad (3.95.)$$

(3.89.) egyenletből:

$$c_{n+1} = (b_n + b_{n+1})h_n + c_n \qquad (3.96.)$$

Továbbá zsenialitásunknak köszönhetően tudjuk azt is, hogy $c_{n+1} = S_n'(x_{n+1}) = q$, így pedig:

$$c_n = q - (b_n + b_{n+1})h_n \qquad (3.97.)$$

Ha (3.90.) egyenletben $i = n$ kifejezést használunk, akkor:

$$c_n = \frac{d_{n+1} - d_n}{h_n} - \frac{1}{3}(2b_n + b_{n+1})h_n \qquad (3.98.)$$

Ebbe visszahelyettesítjük (3.97.) egyenletet, amivel:

$$(2b_{n+1} + b_n)h_n = -\frac{3(d_{n+1} - d_n)}{h_n} + 3q \qquad (3.99.)$$

Már csak összekombináljuk (3.99.), (3.95.) és (3.93.) egyenleteket egy $n+1$ egyenletből álló $n+1$ ismeretlennel $(b_i,\ i=1,2,\dots,n+1)$ rendelkező egyenletrendszerré. Így b_i kifejezhető a következő egyenletrendszer megoldásával:

$$(2b_1+b_2)h_1=\frac{3(d_2-d_1)}{h_1}-3p$$

$$b_{i-1}h_{i-1}+2b_i(h_i+h_{i-1})+b_{i+1}h_i=\frac{3(d_{i+1}-d_i)}{h_i}-\frac{3(d_i-d_{i-1})}{h_{i-1}} \qquad (3.100.)$$

ahol $i=2,3,\dots,n$

$$(2b_{n+1}+b_n)h_n=-\frac{3(d_{n+1}-d_n)}{h_n}+3q$$

Ezzel létrehoztunk egy tridiagonális rendszert (olyan együttható mátrix, melyben csak a főátlóban és a mellette található két átló mentén helyezkednek el nem nulla elemek) egy egyértelmű megoldással. Amint meghatározzuk vele b_i értékeket, (3.90.) egyenlettel megkereshetjük c_i-ket is, végül pedig (3.87.) egyenlettel hozzájutunk a hőn áhított a_i együtthatókhoz. És készen is vagyunk ilyen egyszerű volt! Az átverés az egészben, hogy habár ez most elég bonyolultnak és körülményesnek tűnik, nagyobb elemszámra sokkal jobb, pontosabb és szignifikánsabb gyorsabb eredményt kínál, mint a többi interpoláló eljárás.

3.12. Példa

Feladat:

Oldjuk meg a 3.11 Példában található interpolációt köbös spline-okkal rögzített peremfeltételekkel oly módon, hogy a rögzített peremfeltételek:

$$p=-1; q=1$$

Megoldás:

Mivel 4 bemenő pontunk van, ezért $n=3$, vagyis 3 darab köbös polinomot kell felírni a következő alakban:

$$S_i(x)=a_i(x-x_i)^3+b_i(x-x_i)^2+c_i(x-x_i)+d_i,\quad i=1,2,3$$

Az előző levezetést használva először a b_i értékeket határozzuk meg a következő egyenletek alapján:

$$(2b_1 + b_2)h_1 = \frac{3(d_2 - d_1)}{h_1} - 3p$$

$$b_1h_1 + 2b_2(h_2 + h_1) + b_3h_2 = \frac{3(d_3 - d_2)}{h_2} - \frac{3(d_2 - d_1)}{h_1}$$

$$b_2h_2 + 2b_3(h_3 + h_2) + b_4h_3 = \frac{3(d_4 - d_3)}{h_3} - \frac{3(d_3 - d_2)}{h_2}$$

$$(2b_4 + b_3)h_3 = -\frac{3(d_4 - d_3)}{h_3} + 3q$$

Mivel d_i értékei egyszerűen a bemeneti pontok lesznek, így:

$$d_1 = 5; \quad d_2 = 2{,}3; \quad d_3 = 5{,}1; \quad d_4 = 1{,}5$$

Hasonló elv alapján $h_1 = 1; h_2 = 2; h_3 = 2{,}5$. Ha behelyettesítünk, észben tartva, hogy $p = -1$ és $q = 1$, akkor a rendszer redukálva felírható:

$$\begin{bmatrix} 2 & 1 & 0 & 0 \\ 1 & 6 & 2 & 0 \\ 0 & 2 & 9 & 2{,}5 \\ 0 & 0 & 2{,}5 & 5 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix} = \begin{bmatrix} -5{,}1 \\ 12{,}3 \\ -8{,}52 \\ 7{,}32 \end{bmatrix}$$

ami egy tridiagonális mátrix, megoldva az egyenletrendszert pedig kijön:

$$b_1 = -4{,}3551; \quad b_2 = 3{,}6103; \quad b_3 = -2{,}5033; \quad b_4 = 2{,}7157$$

Következőnek a c_i értékeket határozzuk meg:

$$c_1 = \frac{d_2 - d_1}{h_1} - \frac{1}{3}(2b_1 + b_2)h_1 = -1$$

$$c_2 = \frac{d_3 - d_2}{h_2} - \frac{1}{3}(2b_2 + b_3)h_2 = -1{,}7449$$

$$c_3 = \frac{d_4 - d_3}{h_3} - \frac{1}{3}(2b_3 + b_4)h_3 = 0{,}4691$$

Végezetül pedig jöjjenek az a_i értékek:

$$a_1 = \frac{b_2 - b_1}{3h_1} = 2{,}6551$$

$$a_2 = \frac{b_3 - b_2}{3h_2} = -1{,}0189$$

$$a_3 = \frac{b_4 - b_3}{3h_3} = 0{,}6959$$

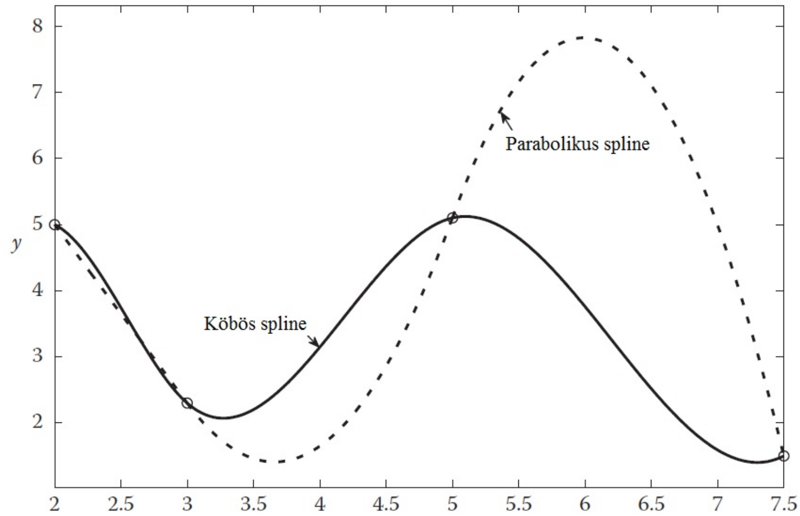
Ezek alapján a 3 köbös spline a következőképpen írható fel:

$$S_1(x) = 2{,}6551(x-2)^3 - 4{,}3551(x-2)^2 - (x-2) + 5, \quad 2 \le x \le 3$$

$$S_2(x) = -1{,}0189(x-3)^3 + 3{,}6103(x-3)^2 - 1{,}7449(x-3) + 2{,}3, \quad 3 \le x \le 5$$

$S_3(x) = 0{,}6959(x-5)^3 - 2{,}5033(x-5)^2 + 0{,}4691(x-5) + 5{,}1, \;\; 5 \le x \le 7{,}5$

Ha vetünk egy pillantást a bemeneti pontokra, és hogy hogyan viselkedik az azokra illesztett parabolikus és köbös spline (lásd ábra alul), hamar belátható, hogy a köbös spline-ok jóval simábban, kisebb kiugrásokkal tudnak illeszkedni, ezzel pontosabb megoldást biztosítva.



3.3.3.5 Köbös spline-ok szerkesztése szabad peremfeltételekkel

Ahogy ezt korábban tisztáztuk, a szabad peremfeltétel nem azok teljes hiányát jelenti, hanem: $S_1''(x_1) = 0$ és $S_n''(x_{n+1}) = 0$ feltételeket. Tudva azt, hogy $S_1''(x) = 6a_1(x - x_1) + 2b_1$, az első feltétel szerint $b_1 = 0$. A levezetések alapján pedig $b_{n+1} = \frac{1}{2} S_n''(x_{n+1})$, vagyis a második feltétel a $b_{n+1} = 0$ lesz. Felhasználva a (3.93.) egyenletet és a két feltételünket egy $n+1$ egyenletből álló, $n+1$ ismeretlent tartalmazó egyenletrendszert írhatunk fel, mely megoldható b_i értékekre, $i = 2,3, \dots, n$ értékekre nézve:

$$b_1 = 0$$

$$b_{i-1}h_{i-1} + 2b_i(h_i + h_{i-1}) + b_{i+1}h_i = \frac{3(d_{i+1} - d_i)}{h_i} - \frac{3(d_i - d_{i-1})}{h_{i-1}} \qquad (3.101.)$$

$$b_{n+1} = 0$$

Ha kiszámoltuk b_i értékeit, a többi hiányzó együttható ugyanazon módon számítható ki, mint rögzített peremfeltételek esetén: $d_i \; (i = 1,2, \dots, n+1)$

értékei a bemeneti pontokkal egyenlők, h_i $(i = 1,2, \dots, n)$ értékeit az osztásköz határozza meg, b_i-k a (3.101.) egyenletből adódnak, c_i-k megkaphatók a (3.90.) egyenletből, a_i-k pedig (3.87.) egyenletből kifejezve:

$$a_i = \frac{b_{i+1} - b_i}{3h_i}, \quad i = 1,2, \dots, n \tag{3.102.}$$

3.13. Példa

Feladat:

A jó ismerős pontokra a 3.11 példából illesszünk szabad peremfeltételekkel rendelkező, köbös spline-okat!

Megoldás:

Mivel szabad peremfeltételekkel dolgozunk, így $b_1 = 0, b_4 = 0$. Következésképpen, (3.101.) egyenlet sokkal barátságosabb lesz:

$$b_1 = 0$$

$$6b_2 + 2b_3 = 12{,}3 \rightarrow b_2 = 2{,}5548$$

$$2b_2 + 9b_3 = -8{,}52 \rightarrow b_3 = -1{,}5144$$

$$b_4 = 0$$

Ezután kiszámoljuk c_i értékeit:

$$c_1 = \frac{d_2 - d_1}{h_1} - \frac{1}{3}(2b_1 + b_2)h_1 = -3{,}5516$$

$$c_2 = \frac{d_3 - d_2}{h_2} - \frac{1}{3}(2b_2 + b_3)h_2 = -0{,}9968$$

$$c_3 = \frac{d_4 - d_3}{h_3} - \frac{1}{3}(2b_3 + b_4)h_3 = 1{,}0840$$

Végezetül, a_i értékeit határozzuk meg:

$$a_1 = \frac{b_2 - b_1}{3h_1} = 0{,}8516$$

$$a_2 = \frac{b_3 - b_2}{3h_2} = -0{,}6782$$

$$a_3 = \frac{b_4 - b_3}{3h_3} = 0{,}2019$$

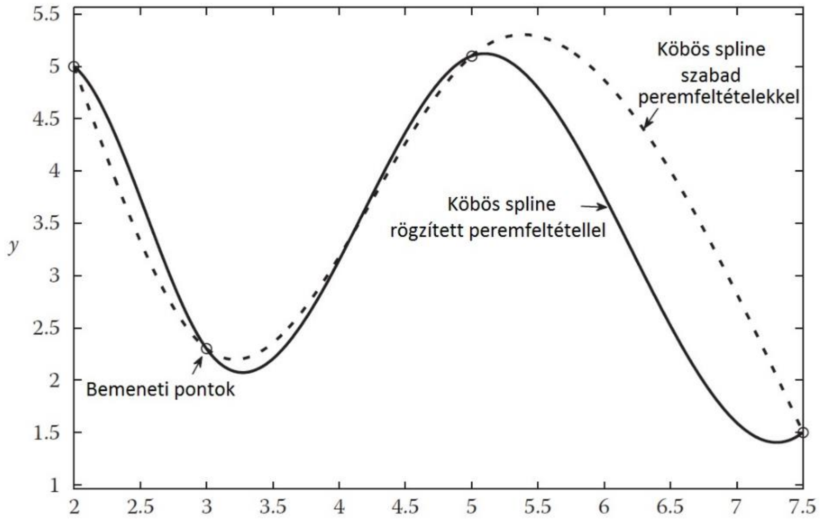
Így előáll a 3 köbös spline:

$$S_1(x) = 0{,}8516(x-2)^3 - 3{,}5516(x-2) + 5; \quad 2 \le x \le 3$$

$$S_2(x) = -0{,}6782(x-3)^3 + 2{,}5548(x-3)^2 - 0{,}9968(x-3) + 2{,}3; \quad 3 \le x \le 5$$

$$S_3(x) = 0{,}2019(x-5)^3 - 1{,}5144(x-5)^2 + 1{,}0840(x-5) + 5{,}1; \quad 5 \le x \le 7{,}5$$

A kötött és szabad peremfeltételekkel meghatározott köbös spline-ok közötti különbséget az alábbi ábra mutatja:



3.14. Példa

Nézzünk pár példát az interpolációs eljárásokra Excel Makróban kidolgozva!

Az elmélet már ismert, tisztázott, ezen gyakorlat során azt nézzük meg, hogy mily módon lehet megoldani a lineáris interpolációt VB használatával:

```
Option Explicit

Function InterpL(keresett_ertek, ismert_x, ismert_y)

Dim pointer As Integer
Dim X0 As Double, Y0 As Double, X1 As Double, Y1 As Double

pointer = Application.Match(keresett_ertek, ismert_x, 1)
X0 = ismert_x(pointer)
Y0 = ismert_y(pointer)
```

```
X1 = ismert_x(pointer + 1)
Y1 = ismert_y(pointer + 1)
InterpL = Y0 + (keresett_ertek - X0) * (Y1 - Y0) / (X1 - X0)
End Function
```

Ez eddig nem is volt nehéz, nem? Tiszta, barátságos, egyszerű, teszi a dolgát, imádnivaló. Hogyan működik egy harmadfokú interpoláció kódja?

```
Function InterpC(keresett_ertek, ismert_x, ismert_y)

'Köbös interpolációt hajtunk végre.
'Fontos: az x-eknek növekvő sorrendben kell lenni!

Dim row As Integer
Dim i As Integer, j As Integer
Dim Q As Double, Y As Double

row = Application.Match(keresett_ertek, ismert_x, 1)
  If row < 2 Then row = 2
  If row > ismert_x.Count - 2 Then row = ismert_x.Count - 2

For i = row - 1 To row + 2
  Q = 1
For j = row - 1 To row + 2
  If i <> j Then Q = Q * (keresett_ertek - ismert_x(j)) / (ismert_x(i) - ismert_x(j))
Next j
  Y = Y + Q * ismert_y(i)
Next i
InterpC = Y

End Function
```

Gyorsan próbáljuk is ki ezeket a függvényeket!

Adottak egy hűtőfolyadék adatai (A, B, C és D oszlop). F oszlopba határozzuk meg azt, hogy mi lesz a folyadék fagyáspontja 33,3%-os tömegszázalékú folyadékra vetítve!

Bemelegítésként próbáljuk ki a TREND függvényt. Ha ez megvan, utána pedig teszteljük a két függvényünket!

	A	B	C	D	E	F	G	H	I
1	Fagyási és forrási pontjai egy folyadéknak								
2	Tömeg% Etilén glikol	Fagyáspont °F	Forráspont °F (1 atm)	Törésindex 22 °C-on		Inderpolálandó xi érték: Tömeg% Etilén glikol			
3	0	32	212	1,3328		33,3			
4	5	29,4	213	1,3378					
5	10	26,2	214	1,3428		Trend függvény használatával (lineáris eset):			
6	15	22,2	215	1,3478		2,45			
7	20	17,9	216	1,353					
8	21	16,8	216	1,354		InterpL függvény használatával:			
9	22	15,9	216	1,3551		2,45			
10	23	14,9	217	1,3561					
11	24	13,7	217	1,3572		InterpC függvény használatával:			
12	25	12,7	218	1,3582		2,46645			
13	26	11,4	218	1,3593					
14	27	10,4	218	1,3603					
15	28	9,2	219	1,3614					
16	29	8	219	1,3624					
17	30	6,7	220	1,3635					
18	31	5,4	220	1,3646					
19	32	4,2	220	1,3656					
20	33	2,9	220	1,3667					
21	34	1,4	220	1,3678					
22	35	-0,2	221	1,3688					

Most próbáljunk meg egy spline függvényt írni!

```
Function spline(periodcol As Range, ratecol As Range, x As Range)

Dim period_count As Integer
Dim rate_count As Integer
period_count = periodcol.Rows.Count
rate_count = ratecol.Rows.Count
If period_count <> rate_count Then
   spline = "Error: Range count dos not match"
   GoTo endnow
End If

ReDim xin(period_count) As Single
ReDim yin(period_count) As Single
```

```
Dim c As Integer
For c = 1 To period_count
xin(c) = periodcol(c)
yin(c) = ratecol(c)
Next c
Dim n As Integer
Dim i, k As Integer
Dim p, qn, sig, un As Single
ReDim u(period_count - 1) As Single
ReDim yt(period_count) As Single
n = period_count
yt(1) = 0
u(1) = 0
For i = 2 To n - 1
   sig = (xin(i) - xin(i - 1)) / (xin(i + 1) - xin(i - 1))
   p = sig * yt(i - 1) + 2
   yt(i) = (sig - 1) / p
   u(i) = (yin(i + 1) - yin(i)) / (xin(i + 1) - xin(i)) - (yin(i) - yin(i - 1)) / (xin(i) - xin(i - 1))
   u(i) = (6 * u(i) / (xin(i + 1) - xin(i - 1)) - sig * u(i - 1)) / p

   Next i

qn = 0
un = 0
yt(n) = (un - qn * u(n - 1)) / (qn * yt(n - 1) + 1)
For k = n - 1 To 1 Step -1
   yt(k) = yt(k) * yt(k + 1) + u(k)
Next k
Dim klo, khi As Integer
Dim h, b, a As Single
klo = 1
khi = n
Do
k = khi - klo
If xin(k) > x Then
khi = k
Else
klo = k
End If
```

```
k = khi - klo
Loop While k > 1
h = xin(khi) - xin(klo)
a = (xin(khi) - x) / h
b = (x - xin(klo)) / h
Y = a * yin(klo) + b * yin(khi) + ((a ^ 3 - a) * yt(klo) + (b ^ 3 - b) * yt(khi)) * (h ^ 2) /
6

spline = Y
endnow:
End Function
```

Az előző adatokra köbös spline-nal való közelítés a következő értéket adja:

	A	B	C	D	E	F	G	H	I
1	Fagyási és forrási pontjai egy folyadéknak								
2	Tömeg% Etilén glikol	Fagyáspont °F	Forráspont °F (1 atm)	Törésindex 22 °C-on		Inderpolálandó xi érték: Tömeg% Etilén glikol			
3	0	32	212	1,3328		33,3			
4	5	29,4	213	1,3378					
5	10	26,2	214	1,3428		Trend függvény használatával (lineáris eset):			
6	15	22,2	215	1,3478		2,45			
7	20	17,9	216	1,353					
8	21	16,8	216	1,354		InterpL függvény használatával:			
9	22	15,9	216	1,3551		2,45			
10	23	14,9	217	1,3561					
11	24	13,7	217	1,3572		InterpC függvény használatával:			
12	25	12,7	218	1,3582		2,46645			
13	26	11,4	218	1,3593					
14	27	10,4	218	1,3603		Spline			
15	28	9,2	219	1,3614		2,468572			
16	29	8	219	1,3624					
17	30	6,7	220	1,3635					
18	31	5,4	220	1,3646					
19	32	4,2	220	1,3656					
20	33	2,9	220	1,3667					
21	34	1,4	220	1,3678					
22	35	-0,2	221	1,3688					

3.15. Példa: Lagrange Interpoláció

Erre azért nem volt túl barátságos a táblázatunk, ha emlékszünk. Itt kezd el igazán hasznos lenni a VB, és a programozási alapismeretek, mert a Lagrange interpoláció sokkal egyszerűbben kivitelezhető programozott formájában.

Tegyük fel, hogy adottak a következő adatok:

	A	B	C	D	E
1	Lagrange Interpoláció				
2					
3	Adatok			Fokszám	3
4	x	y		xi	10
5	13	4755			
6	77	3940		f(x) =	
7	5	3090			
8	3	2310			
9	1	800			

Szeretnénk meghatározni valamely (esetünkben xi = 10) pontban a függvény értékét úgy, hogy a Lagrange polinom fokszámát mi határozzuk meg, majd irassuk ki egy Message Box-ban az eredményt!

```
Option Explicit

Sub Lagrange_Interpolacio()

Dim n, i, order As Integer
Dim x(10), Y(10), xi As Double

'x adatok kiválasztása
For i = 0 To 10
   x(i) = Range("A" & i + 5)
Next

'y adatok kiválasztása
For i = 0 To 10
   Y(i) = Range("B" & i + 5)
Next

'xi kiválasztása
xi = Range("E4")

'Lagrange fokozatának kiválasztása
order = Range("E3")

'a kimenetet kiírjuk
```

```
MsgBox "A függvény becsült értéke: " & Lagrange(x, Y, order, xi)

End Sub
```

```
Function Lagrange(x, Y, order, xi)

Dim i, j As Integer
Dim sum, prod As Double

'sum értékének inicializálása
sum = 0

For i = 0 To order
prod = Y(i)

For j = 0 To order

If i <> j Then
   prod = prod * (xi - x(j)) / (x(i) - x(j))
End If

Next j
sum = sum + prod
Next i

Lagrange = sum

End Function
```

Figyeljük meg, hogy hogyan változik a becsült függvény értéke, ha változtatjuk az interpoláció fokszámát!

3.16. Példa: Kétváltozós függvények Interpolációja

Végül nézzünk egy kissé komplexebb példát: mi a helyzet kétváltozós függvények interpolációjával? Ilyenkor, habár kivitelezhető a hagyományos módon is az eljárás, csinálja azt az, akinek két anyja van. Ekkor a saját függvényen kívül már más nem igazán jöhet szóba:

```
Option Explicit
Option Base 1

Function InterpC2(keresett_x, keresett_y, ismert_x, ismert_y, ismert_z)

Dim M As Integer, N As Integer
Dim R As Integer, C As Integer
Dim XX(4) As Double, YY(4) As Double, ZZ(4) As Double, ZInterp(4) As Double

R = Application.Match(keresett_x, ismert_x, 1)
C = Application.Match(keresett_y, ismert_y, 1)
If R < 2 Then R = 2
If R > ismert_x.Count - 2 Then R = ismert_x.Count - 2
If C < 2 Then C = 2
If C > ismert_y.Count - 2 Then C = ismert_y.Count - 2

For N = 1 To 4
XX(N) = ismert_x(R + N - 2)
If ismert_y(C + 2) > ismert_y(C - 1) Then
   For M = 1 To 4
      YY(M) = ismert_y(C + M - 2)
      If ismert_z(R + N - 2, C + M - 2) = "" Then InterpC2 = CVErr(xlErrNA): Exit
Function
      ZZ(M) = ismert_z(R + N - 2, C + M - 2)
   Next M
Else
   For M = 1 To 4
      YY(M) = ismert_y(C - M + 3)
      If ismert_z(R + N - 2, C - M + 3) = "" Then InterpC2 = CVErr(xlErrNA): Exit
Function
      ZZ(M) = ismert_z(R + N - 2, C - M + 3)
   Next M
End If
ZInterp(N) = CI(keresett_y, YY, ZZ)
'Ez az array a Z értékek interpolációja keresett_y helyeken
Next N

InterpC2 = CI(keresett_x, XX, ZInterp)
End Function
```

```
Private Function CI(keresett_ertek, ismert_x, ismert_y)

Dim i As Integer, j As Integer
Dim Q As Double, Y As Double

For i = 1 To 4
  Q = 1
For j = 1 To 4
  If i <> j Then Q = Q * (keresett_ertek - ismert_x(j)) / (ismert_x(i) - ismert_x(j))
Next j
  Y = Y + Q * ismert_y(i)
Next i
CI = Y
End Function
```

Próbáljuk ki a következő táblázaton a működését:

	A	B	C	D	E	F	G	H	I
1		**Hűtőfolyadék viszkozitásának változása a hőmérséklet és összetétel függvényében**							
2		Térfogatszázalék - etilén - glikol							
3	Hőmérséklet °F	20	30	40	50	60	70	80	90
4	0			13,76	19,34	30,08	45,58	65,04	107,77
5	10		6,83	10,13	14,26	22,06	33,31	46,89	71,87
6	20	3,9	5,38	7,74	10,85	16,56	24,79	34,48	49,94
7	30	3,24	4,33	6,09	8,48	12,68	18,77	25,84	35,91
8	40	2,59	3,54	4,91	6,77	9,9	14,45	19,71	26,59
9	50	2,18	2,95	4,04	5,5	7,85	11,31	15,29	20,18
10	60	1,86	2,49	3,38	4,55	6,33	8,97	12,05	15,65
11	70	1,61	2,13	2,87	3,81	5,17	7,22	9,62	12,37
12	80	1,41	1,84	2,46	3,23	4,28	5,88	7,79	9,93
13	90	1,24	1,6	2,13	2,76	3,58	4,85	6,38	8,1
14	100	1,11	1,41	1,87	2,39	3,03	4,04	5,28	6,68
15	110	0,99	1,25	1,64	2,08	2,58	3,4	4,41	5,58

Ha mindent jól csináltunk, akkor a függvényünk a következőképpen dolgozik:

K	L	M
Használt hőmérséklet	Használt térfogatszázalék	Interpolált Viszkozitás
95	74,5	5,013713867

4 Numerikus differenciálás

A mérnöki problémák során gyakran előfordul, hogy egy adott függvénynek, vagy pedig egy adott bemeneti ponthalmazra illesztett függvénynek meg kell állapítani az első, második, stb. deriváltját. Habár itt egész sokáig sorolhatnánk, hogy hányadik deriváltat számítsuk ki, gyakorlatban igen ritka, hogy a második derivált értékénél mélyebbre kelljen ásnunk. A deriválás előnye az integrálással szemben, hogy viszonylag kevés szabállyal, még analitikusan is könnyen kivitelezhető egyszerűbb függvényekre, polinomokra. Viszont összetett függvények esetén nem biztos, hogy a hagyományos módszer a legcélravezetőbb (ráadásul az is előfordulhat, hogy ily módon nem is lehet megoldani a problémát), hiszen numerikus módszerekkel hasonló pontosság biztosítható jóval alacsonyabb műveletszámmal is. A numerikus eljárások során a függvényünket diszkretizáljuk, vagyis több pontot hozunk létre, ahol a függvény értékének a segítségével meg tudjuk határozni a függvény deriváltjának értékét egy bizonyos keresett pontban. Ez azért is célszerű megközelítés, hiszen ha éppen bemenő pontok állnak rendelkezésre és nem is egy függvény, akkor már készen is vagyunk; megspóroltunk egy lépést magunknak. Ha a pontok előálltak, két lehetőségünk van a derivált számítására: véges differenciák alkalmazásával, vagy pedig egy könnyen illeszthető függvény (előző fejezet) megkeresésével és annak deriválásával. Természetesen ez a fejezet az első esetet fogja taglalni.

Habár nemsokára látni fogjuk, hogy számos módszer áll rendelkezésre, azok felépítése, levezetése igen hasonló, így szerencsére ez a fejezet jóval rövidebb lesz, mint az előző volt.

A véges differencia formulák lényege, hogy egy adott x_i pontban a környezetében lévő függvény értékek segítségével próbálják meg közelíteni a függvény első, második, stb. deriváltjának értékét. A különböző rendű deriváltak becsléséhez pedig a Taylor-sorba fejtést használjuk fel. A fejezetben 4 példát nézünk meg a véges differencia formulák meghatározására, utána pedig belátjuk/elhisszük, hogy a többi formula is ugyanezen módon származtatható.

A formulák különbözősége abból adódik, hogy hány környező függvényértéket használnak fel a derivált meghatározásához. Ezek a függvényértékek – pontok eshetnek jobbra, balra vagy lehetnek mindkét oldalán a vizsgált x_i ponttól, ettől függően beszélhetünk előre, közép vagy hátra differenciáról (ez tulajdonképpen egy elég izzadságszagú magyarítása az angol *forward, central, backward* kifejezéseknek).

4.1 Két pontos hátra differenciál [3]

Az $f(x_{i-1})$ értékét Taylor-sorba fejtéssel tudjuk megbecsülni x_i pontban, ha $h = x_i - x_{i-1}$ osztásközt alkalmazunk:

$$f(x_{i-1}) = f(x_i) - hf'(x_i) + \frac{1}{2!}h^2 f''(x_i) - \frac{1}{3!}h^3 f'''(x_i) + \cdots \tag{4.1.}$$

A lineáris tagokat meghagyva:

$$f(x_{i-1}) = f(x_i) - hf'(x_i) + \frac{1}{2!}h^2 f''(\zeta) \tag{4.2.}$$

ahol a 3. tag lesz a visszamaradó tag, ahol $x_{i-1} \leq \zeta \leq x_i$. Ha megoldjuk (4.2.) egyenletet:

$$f'(x_i) = \frac{f(x_i) - f(x_{i-1})}{h} + \frac{1}{2!}hf''(\zeta) \tag{4.3.}$$

ahol a második tag a lekerekítési hibát fogja adni. Ha nagyvonalúan ezt a tagot elhanyagoljuk, akkor megkaphatjuk az első derivált becsült értékét. Persze szem előtt kell tartani, hogy a lekerekítési hiba lineárisan függ h osztásköztől, pontosabban annak nagyságrendjétől, ezért felírhatjuk, mint $O(h)$, így pedig megkapjuk a két pontos, hátra differenciál képletét:

$$f'(x_i) = \frac{f(x_i) - f(x_{i-1})}{h} + O(h) \tag{4.4.}$$

Mivel ζ értékét nem ismerjük pontosan, így $O(h)$ értéke sem ismert. Azt viszont tudjuk, hogy h minél kisebb, $O(h)$ értéke úgy válik elhanyagolhatóvá.

4.2 Két pontos előre differenciál

Ahogy az előbb megjósoltuk, ugyanazon elv szerint fogunk haladni: vagyis az $f(x_{i+1})$ értékét továbbra is Taylor- sorba fejtéssel fogjuk meghatározni x_i pontból kiindulva, $h = x_{i+1} - x_i$ osztásközzel:

$$f(x_{i+1}) = f(x_i) - hf'(x_i) + \frac{1}{2!}h^2 f''(x_i) - \frac{1}{3!}h^3 f'''(x_i) + \cdots \tag{4.5.}$$

A lineáris tagok:

$$f(x_{i+1}) = f(x_i) - hf'(x_i) + \frac{1}{2!}h^2 f''(\zeta) \tag{4.6.}$$

ahol a 3. tag ismét a visszamaradó tag, illetve $x_i \leq \zeta \leq x_{i+1}$. Ha megoldjuk az egyenletet:

$$f'(x_i) = \frac{f(x_{i+1}) - f(x_i)}{h} + \frac{1}{2!}hf''(\zeta) \qquad (4.7.)$$

A második tag újra a lekerekítési hibát tartalmazza. Hasonló elhanyagolással élve, mint az előbb, a végső formula:

$$f'(x_i) = \frac{f(x_{i+1}) - f(x_i)}{h} + O(h) \qquad (4.8.)$$

4.3 Két pontos közép differenciál

Ebben az esetben továbbra is a Taylor-sorba fejtést fogjuk használni, viszont most már szükségünk lesz a négyzetes tagra is:

$$f(x_{i-1}) = f(x_i) - hf'(x_i) + \frac{1}{2!}h^2 f''(x_i) - \frac{1}{3!}h^3 f'''(\zeta); \quad x_{i-1} \leq \zeta \leq x_i \qquad (4.9.)$$

illetve:

$$f(x_{i+1}) = f(x_i) + hf'(x_i) + \frac{1}{2!}h^2 f''(x_i) + \frac{1}{3!}h^3 f'''(\varphi); \quad x_{i-1} \leq \varphi \leq x_i \qquad (4.10.)$$

Ha a második egyenletből kivonjuk az elsőt, akkor a következőt kapjuk:

$$f(x_{i+1}) - f(x_{i-1}) = 2hf'(x_i) + \frac{1}{3!}h^3[f'''(\zeta) + f'''(\varphi)] \qquad (4.11.)$$

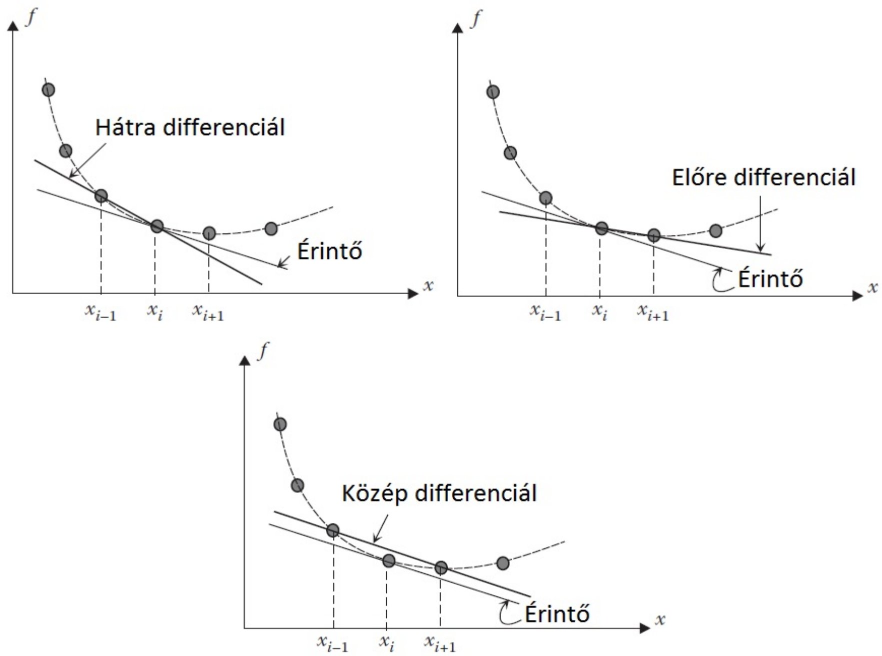
Ebből könnyedén kifejezzük $f'(x_i)$ értékét:

$$f'(x_i) = \frac{f(x_{i+1}) - f(x_{i-1})}{2h} + O(h^2) \qquad (4.12.)$$

Mire is volt jó ez az egész? Látható, hogy a lekerekítési hiba itt már az osztásköz négyzetétől függ, így annak értéke jóval alacsonyabb lesz, ergo jóval pontosabb közelítést kapunk, mint az előre vagy hátra differenciál formulák esetében.

Tegyük fel, hogy adottak $x_1, x_2, \dots, x_n$ pontok. Ekkor az eddig megismert 3 módszer közül a kétpontos hátra differenciál formula nem használható x_1 pontban, hiszen nem áll rendelkezésre x_0, viszont az összes többi pontban használható $O(h)$ pontossággal. A kétpontos előre differenciál képletével fordított a helyzet: ez a teljes tartományon használható $O(h)$ pontossággal, kivéve x_n pontban, hiszen x_{n+1} nem adott. A kétpontos közép differenciál pedig a kezdő- és végpontban (x_1 és x_n) nem használható – hasonló indokok miatt –

viszont a tartomány többi részén igen, méghozzá $O(h^2)$ pontossággal. Ezt érdemes szem előtt tartani egy ilyen probléma kezelésekor.



11. ábra: Két pontos differenciál formulák szemléltetése első derivált értékének közelítésére

4.4 Három pontos hátra differenciál

Az első derivált három ponttal való közelítése során ugyanazon elv szerint mozgunk, mit eddig: először Taylor-sorba fejtéssel kifejezzük $f(x_{i-1})$ értékét x_i pontban:

$$f(x_{i-1}) = f(x_i) - hf'(x_i) + \frac{1}{2!}h^2 f''(x_i) - \frac{1}{3!}h^3 f'''(\zeta); \quad x_{i-1} \le \zeta \le x_i \qquad (4.13.)$$

Ezután $f(x_{i-2})$ értéket közelítjük x_i pontból:

$$f(x_{i-2}) = f(x_i) - (2h)f'(x_i) + \frac{1}{2!}(2h)^2 f''(x_i) - \frac{1}{3!}(2h)^3 f'''(\varphi); \quad x_{i-2} \le \varphi \le x_{i-1} \qquad (4.14.)$$

Ha (4.13.) egyenlet 4-szeresét kivonjuk (4.14.) egyenletből, akkor a következő kifejezést kapjuk:

$$f(x_{i-2}) - 4f(x_{i-1}) = -3f(x_i) + hf'(x_i) + \frac{4}{3!}h^3 f'''(\zeta) - \frac{8}{3!}h^3 f'''(\varphi) \qquad (4.15.)$$

Ezután kifejezzük $f'(x_i)$ értékét:

$$f'(x_i) = \frac{f(x_{i-2}) - 4f(x_{i-1}) + 3f(x_i)}{2h} - \frac{1}{3}h^2 f'''(\zeta) + \frac{2}{3}h^2 f'''(\varphi) \qquad (4.16.)$$

Mivel a lekerekítési hiba továbbra is négyzetesen függ az osztásköztől, így tovább egyszerűsítve a formula:

$$f'(x_i) = \frac{f(x_{i-2}) - 4f(x_{i-1}) + 3f(x_i)}{2h} + O(h^2) \qquad (4.17.)$$

Vagyis a három pontos hátra differenciál formula az x_i helyen vett első derivált értékét a függvény x_i, x_{i-1} és x_{i-2} helyen vett függvény értékekkel tudja közelíteni.

4.5 Numerikus differenciálási formulák

A második, harmadik deriváltak értékei hasonló módon származtathatók a függvény értékeiből, x_i kiindulási ponttal, Taylor-sorba fejtéssel, így ezek itt most nem kerülnek kifejtésre, levezetésre. A leggyakrabban használt differenciálási formulákat a következő táblázat tartalmazza.

Első derivált, 2 pont használatával	
„Előre” differenciál	$y_i' = \frac{y_{i+1} - y_i}{h}$
Közép differenciál	$y_i' = \frac{y_{i+1} - y_{i-1}}{2h}$
„Hátra” differenciál	$y_i' = \frac{y_i - y_{i-1}}{h}$
Első derivált, 3 pont használatával	
„Előre” differenciál	$y_i' = \frac{-y_{i+2} + 4y_{i+1} - 3y_i}{2h}$
Első derivált, 4 pont használatával	
Közép differenciál	$y_i' = \frac{-y_{i+2} + 8y_{i+1} - 8y_{i-1} + y_{i-2}}{12h}$
Második derivált, 3 pont használatával	
„Előre” differenciál	$y_i'' = \frac{y_{i+2} - 2y_{i+1} + y_i}{h^2}$

Közép differenciál	$y_i'' = \frac{y_{i+1} - 2y_i + y_{i-1}}{h^2}$
„Hátra” differenciál	$y_i'' = \frac{y_i - 2y_{i-1} + y_{i-2}}{h^2}$
Második derivált, 4 pont használatával	
„Előre” differenciál	$y_i'' = \frac{2y_i - 5y_{i+1} + 4y_{i+2} - y_{i+3}}{h^2}$
Második derivált, 5 pont használatával	
Közép differenciál	$y_i'' = \frac{-y_{i+2} + 16y_{i+1} - 30y_i + 16y_{i-1} - y_{i-2}}{12h^2}$
Harmadik derivált, 4 pont használatával	
„Előre” differenciál	$y_i''' = \frac{y_{i+3} - 3y_{i+2} + 3y_{i+1} - y_i}{h^3}$

4. táblázat: Numerikus differenciálási formulák

4.1. Példa

Feladat:

Van egy harmadfokú polinomunk, aminek sürgősen kellene a meredeksége -10 és -1 között bizonyos pontokban. Legyen adott a következő probléma:

	A	B	C
1	**Numerikus Differenciálás**		
2	**F(x) = tx^3 + ux^2 + vx + w**		
3		t	1
4		u	-3
5		v	-130
6		w	150

Megoldás:

Az első deriváltat pedig számoljuk ki hagyományos, analitikus módon, illetve hozzunk létre egy makrót, ami két pontból előre differenciál módszerrel számolja az első derivált értékét! A makró futtatásánál nekünk kelljen kiválasztani X és Y értékeit, illetve a cél cellákat!

A végeredmény a következőképpen fog kinézni:

	A	B	C	D	E	F	G	H
1	Numerikus Differenciálás							
2	F(x) = tx^3 + ux^2 + vx + w							
3		t	1					
4		u	-3				delta	
5		v	-130				1,00E-08	
6		w	150					
7							Δx/Δy	
8	x	y	Δx	x+Δx	y+Δy	Δy/Δx	Excel képlettel	Makró
9	-10	150	-1,00E+01	1,50E+02	-1,00E-07	-2,30E-05	230,00	230,00
10	-9	348	-9,00E+00	3,48E+02	-9,00E-08	-1,50E-05	167,00	167,00
11	-8	486	-8,00E+00	4,86E+02	-8,00E-08	-8,80E-06	110,00	110,00
12	-7	570	-7,00E+00	5,70E+02	-7,00E-08	-4,13E-06	59,00	59,00
13	-6	606	-6,00E+00	6,06E+02	-6,00E-08	-8,40E-07	14,00	14,00
14	-5	600	-5,00E+00	6,00E+02	-5,00E-08	1,25E-06	-25,00	-25,00
15	-4	558	-4,00E+00	5,58E+02	-4,00E-08	2,32E-06	-58,00	-58,00
16	-3	486	-3,00E+00	4,86E+02	-3,00E-08	2,55E-06	-85,00	-85,00
17	-2	390	-2,00E+00	3,90E+02	-2,00E-08	2,12E-06	-106,00	-106,00
18	-1	276	-1,00E+00	2,76E+02	-1,00E-08	1,21E-06	-121,00	-121,00

Y számítása nem okozhat gondot.

$$\Delta x = x * (1 + delta)$$

$$(x + \Delta) = F(x + \Delta x)$$

$$(y + \Delta) = x * delta$$

$$\frac{\Delta x}{\Delta y} = F'(x)$$

A makróval való megoldása a feladatnak:

```
Option Explicit
Option Base 1

Sub Derivs()
Dim z As Integer, N As Integer
Dim Old_Ys() As Double, New_Ys() As Double, Old_Xs() As Double
Dim Derivs() As Double, increment As Double
Dim known_Xs As Object, known_Ys As Object, cel As Object
increment = 0.00000001

Set known_Ys = Application.InputBox("Válassza ki az Y értékeit!", "Lépés 1/3", , , ,
, , 8)
N = known_Ys.Count
```

```
ReDim Old_Ys(N), New_Ys(N), Old_Xs(N), Derivs(N)
z = 1
For Each cel In known_Ys
   Old_Ys(z) = cel.Value
   z = z + 1
Next cel

Set known_Xs = Application.InputBox("SVálassza ki az X értékeit!", "Lépés 2/3", ,
, , , , 8)
z = 1
For Each cel In known_Xs
   Old_Xs(z) = cel.Value
   cel.Value = Old_Xs(z) * (1 + increment)
   z = z + 1
Next cel
z = 1
For Each cel In known_Ys
   New_Ys(z) = cel.Value
   z = z + 1
Next cel
z = 1
For Each cel In known_Xs
   cel.Value = Old_Xs(z)
   z = z + 1
Next cel

Application.InputBox("Válassza ki, hova tegyem az eredményeket!", "Lépés 3/3",
, , , , , 8).Select
For z = 1 To N
   Derivs(z) = (New_Ys(z) - Old_Ys(z)) / (increment * Old_Xs(z))
   ActiveCell.Offset(z - 1, 0).Value = Derivs(z)
Next

End Sub
```

5 Numerikus integrálás

Az integrálást, mint matematikai eszközt a mérnöki gyakorlatok igen széles spektrumán alkalmazzuk, hiszen a függvény alatti terület számítás képezi a teljesítményszámítás alapját, persze számos egyéb alkalmazása is előfordulhat (gondoljunk csak a hőtani állapotváltozások leírására, nyomatéki görbe meghatározása nyíróerőből, stb.). Szintén gyakori, hogy ott ahol a függvény alatti területre lenne szükség, maga a függvény nem ismert – például egy mérési eredmény kiértékelése során. Így vagy az approximációhoz nyúlunk és a kapott függvényt analitikus módon integráljuk, vagy pedig alapból egy numerikus integrálási módszert választunk, mely lehetőséget kínál az approximáció kihagyására, és in medias res módon kapásból belecsapni az integrálásba. Hogy még inkább meggyőzzek mindenkit a numerikus integrálás gyakorlati hasznosságáról (azon felül, hogy integrálni – szemben a deriválással – nem igazán könnyű), arról se feledkezzünk meg, hogy léteznek szép számmal olyan függvények, melyek a hagyományos úton véve nem integrálhatók.

Emlékeztető: a határozott integrálás problémája alatt a következő képletet értjük:

$$\int_a^b f(x)dx = F(x) \qquad (5.1.)$$

ahol $F(x)$ függvény $f(x)$ primitív függvénye, köztük a kapcsolat: $F'(x) = f(x)$. Ha a és b határoktól eltekintenénk, határozatlan integrálásról beszélnénk, ekkor végtelen sok megoldás létezik, hiszen $F(x) + c, c \in \mathbb{R}$ is $f(x)$ integrálja lesz, mivel a konstans deriváltja mindig 0-ra adódik. A határozott integrálás során az $f(x)$ függvény $[a, b]$ intervallumon értelmezett görbe alatti területét értjük, és ez a mérnöki létünk során elég sokszor lesz különösen fontos, viszont emlékezhetünk rá, hogy olykor milyen kínkeserves is megszülni egy függvény primitív függvényét.

A numerikus integrálás lényege, hogy az analitikus megoldás helyett egy sor diszkrét pont és a vett függvény érték egy adott kombinációjával állítjuk elő, egészen pontosan **<u>közelítjük</u>** az integrál értékét. Vagyis most is diszkretizálnunk kell az értelmezési tartományt, majd valahogyan megszereznünk a függvény értékeit. Ez rendelkezésre állhat alapból egy mérés eredményeképp, vagy pedig kiszámolható, ha a függvény maga ismert. Természetesen az indok, hogy ez miért jó nekünk: mert egyrészt sokkal könnyebb, mint hagyományos módon integrálni (tényleg sokkal!), másrészt jelentősen kevesebb műveletszám szükséges elvégzéséhez, ami főleg egy bonyolultabb szimuláció, vagy nagy adathalmaz esetén igen jól jöhet.

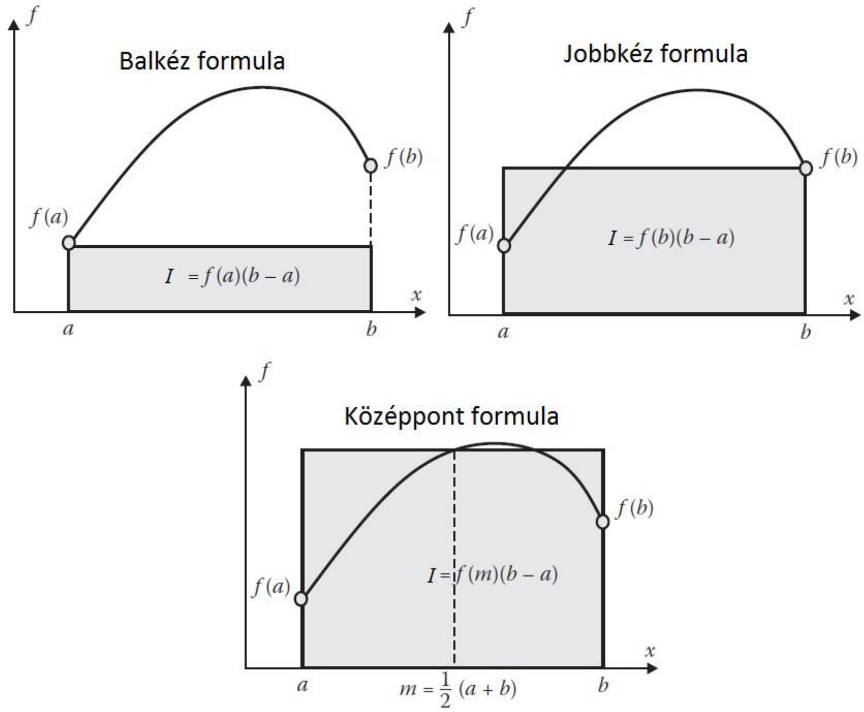
5.1 Newton-Cotes formulák

A leggyakrabban használt formulák a Newton-Cotes formulák, melyek alapvetően két kategóriába sorolhatók: nyitott és zárt alakúba. A különbség a kettő között viszonylag egyszerű: a nyitott formuláknál az intervallum végpontjai nem kerülnek felhasználásra a számítás során, míg a zárt formulák esetében igen. Nyitott formulának minősülnek a téglalapszabály és a Gauss kvadratúra formula, míg zártak lesznek a trapéz- és Simpson szabályok.

A Newton-Cotes formulák egy nagyon egyszerű megközelítésből indulnak ki: ahelyett, hogy a tényleges függvényt integrálnánk, helyettesítsük azt egy nagyon egyszerű polinommal (nulla, első, másod, maximum harmadrendű polinommal), és annak/azoknak az integrálását végezzük el. Ez kétféleképp lehetséges: ha a függvény maga rendelkezésre áll, akkor diszkretizáció segítségével pontokat képezünk, majd interpolációval ráillesztjük a függvényünket, melyet végül integrálunk; ha eleve pontok állnak rendelkezésre, akkor pedig az első lépést ki is hagyhatjuk.

5.1.1 Téglalap szabály

Az egyik legegyszerűbb módszer a téglalap szabály. Ebben az esetben a függvény integrál értékét ($\int_a^b f(x)dx$) egy téglalap segítségével közelítjük a és b végpontok között, ezzel a téglalap szélessége adott is lesz: $b-a$. A kérdés csak az, hogy mily módon határozzuk meg a téglalap magasságát. Erre három megoldást használhatunk: ha az intervallum bal oldali végpontjában vett függvény értéket vesszük alapul, ha a jobb oldali végpontban vett függvény értéket vesszük alapul, vagy ha az intervallum közepén vett függvényértéket. Ezen megkülönböztetés miatt szokás hívni ezeket a módszereket balkéz-, jobbkéz- és középpont módszernek. Habár a módszer maga tényleg könnyű, azért érezhető, hogy pontossága hagy némi kívánnivalót maga után. Ahogy a 12. ábra is szemlélteti, amennyiben az $[a,b]$ intervallum széles, ráadásul a rajta értelmezett $f(x)$ függvényben viszonylag nagy változás történik. A téglalap módszerek mindegyike igen nagy hibával dolgozik, ha a tökéletességre törekszünk, az értelmezési tartomány egyként való kezelése, vagyis a globálisan felírt téglalap módszer maximum nagyságrendben adhat támpontot. Persze pontos eredményként semmiképpen sem ajánlott elfogadni. Ugyanakkor azt is láthatjuk, hogy miképp lehetne ezen a tulajdonságon változtatni: mi történne akkor, ha ugyanezt a feladatot nem csak egyszer, hanem többször is megcsinálnánk? Ezzel jutunk el az ún. összetett téglalap szabályokhoz.



12. ábra: Téglalap szabály formái

Először is az $[a, b]$ intervallumot felosztjuk n darabra ahol:

$$a = x_0 < x_1 < \cdots < x_{n-1} < x_n = b$$

Δx_i szélességű alintervallumokat hozunk létre. A teljes intervallumon értelmezett függvény integrál értéket ezen intervallumokon vett függvényértékek szorzatösszegével fogjuk közelíteni, vagyis:

$$\int_a^b f(x)dx \approx \sum_{i=1}^{n} f(x_i^*)\Delta x_i \qquad (5.2.)$$

Ehhez elsősorban az x_i^* értéket kell megtalálni minden intervallumra, de hát erre találtuk ki a balkéz-, jobbkéz- és középpont szabályt. A három legegyszerűbb módszer a következőképp néz ki felírva:

- balkéz:

$$L_n = \sum_{i=1}^{n} f(x_{i-1})\Delta x_i \qquad (5.3.)$$

- jobbkéz:

$$R_n = \sum_{i=1}^{n} f(x_i)\Delta x_i \qquad (5.4.)$$

- középpont:

$$SM_n = \sum_{i=1}^{n} f\left(\frac{x_{i-1}+x_i}{2}\right)\Delta x_i \qquad (5.5.)$$

Általában véve szeretünk törekedni az egyszerűségre, így gyakran alakítjuk úgy a képletet, hogy az $[a,b]$ intervallum felosztása ekvidisztáns módon történjen, vagyis a részintervallumok szélessége megegyezzen: $h = \frac{b-a}{n}$, ahol n a részintervallumok száma. Ekkor az előző három képlet a következőképpen egyszerűsödik:

- balkéz:

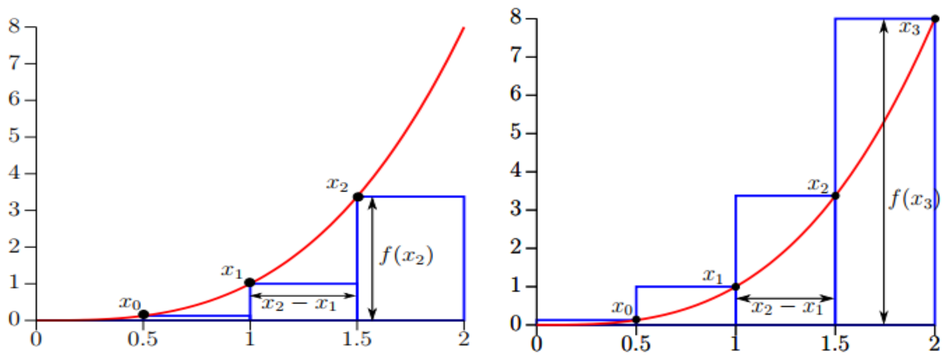
$$L_n = \frac{b-a}{n}\sum_{i=1}^{n} f(x_{i-1}) \qquad (5.6.)$$

- jobbkéz:

$$R_n = \frac{b-a}{n}\sum_{i=1}^{n} f(x_i) \qquad (5.7.)$$

- középpont:

$$SM_n = \frac{b-a}{n}\sum_{i=1}^{n} f\left(\frac{x_{i-1}+x_i}{2}\right) \qquad (5.8.)$$



13. ábra: Példa az összetett balkéz (balra) és jobbkéz (jobbra) szabályra [1]

[1] http://www.inf.u-szeged.hu/~kgelle/sites/default/files/upload/11_numerikus_integralas_0.pdf

5.1. Példa

Feladat:

Vizsgáljuk meg az "$f(x) = 1\text{-}x^2$" függvény alatti területet a= 0 és b = 1,5 között 10, 40 és 100 részintervallumra felosztva!

Megoldás:

A következő oldalon látható Excel táblázatot fogjuk létrehozni a következőképpen:

Az első 10 sorban már nem szabad, hogy legyen bármilyen kihívás, ezért annak részletezésétől eltekintek.

Az intervallumok számának megadásánál definiáljuk n értékét, ami alapján számolnunk kell a h osztásközt, itt az A8-ban megadott értékkel számoltatunk.

Ezután jön az érdemi rész, kezdjük a 10 részintervallumra való felosztással: Az x értékeit megadhatjuk egyszerűen, A16 cellát egyenlővé tesszük B5 cellával. Ettől lefelé viszont teszünk bele egy magyaros huszárcsapást, mert éljen a lustaság. Ahelyett, hogy csak hozzáadogatnánk a h osztásközt A16 cellához, beteszünk egy HA logikai vizsgálatot B17 cellától lefelé:

=HA((A16+B$12)>$B$6;0;(A16+B$12))

Ennek az értelme annyi, hogy nem kell cellákat számolgatni, amikor lefelé elkezdjük másolni a képletet, mert ha eléri az 1,5 értéket, utána automatikusan kinullázza magát, és újrakezdi a számlálást, ezzel könnyebben megtaláljuk, hogy meddig kell húzni a képleteket.

B16 cellától felelé megadjuk $f(x) = 1 - x^2$ függvény képletét. Az A oszloppal együtt ezek fogják az alappontokat képezni.

Ebből számítjuk bal-, jobb- és középmódszer szerint a görbe alatti területet. A bal- és jobbkéz módszer esetében semmi turpisság nincs az ügyben, hiszen C és D oszlopok celláit csak egyenlővé tesszük a mellettük lévő B cellák elemeivel, viszont balkézmódszernél az utolsó, míg jobbkézmódszernél az első cellát hagyjuk üresen. A középszabálynál (E oszlop) szintén üres marad az első sor, majd pedig az előző két függvényérték átlagát vesszük majd, így E17 cellában a következő képletet adjuk meg:

=(B17+B16)/2

Vegül az értékek kiszámítása után összegezzük a függvényértékeket az korában megismert képletek szerint, majd osztjuk őket az osztásközzel, így C28 cella a következő lesz:

=SzUM(C16:C26)*$B12

A többi természetesen hasonlóképpen alakul.

Ugyanezt az eljárást ismételjük meg kétszer, de egyszer 40 intervallumra, egyszer pedig 100 intervallumra osztjuk [a,b]-t.

Numerikus integrálás - balkéz-, jobbkéz érték alapján

Vizsgáljuk meg az "$f(x) = 1-x^2$" függvény alatti területet a= 0 és b = 1,5 között 20 részintervallumra felosztva!

Adott:

a=	0,000
b=	1,500
n=	40,000
h=(b-a)/n=	0,038

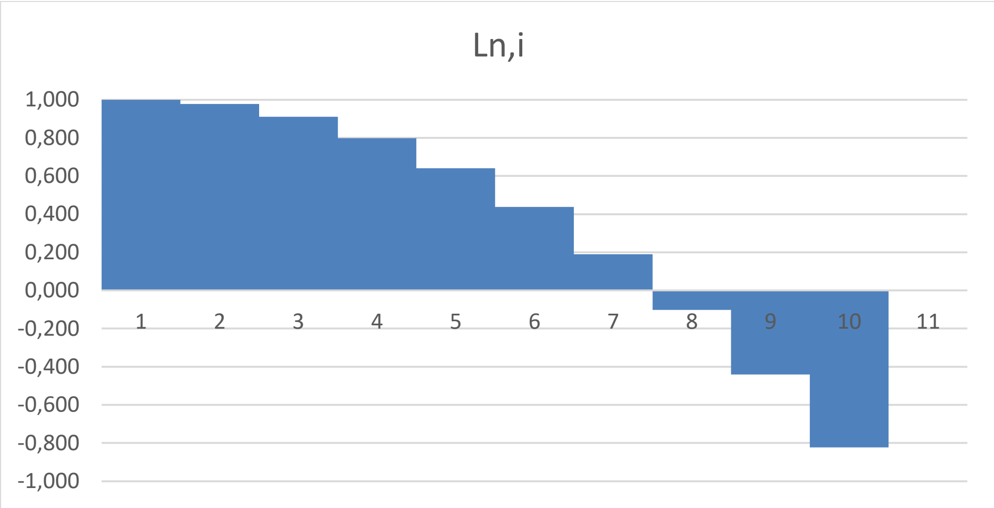
Intervallumok számának megadása

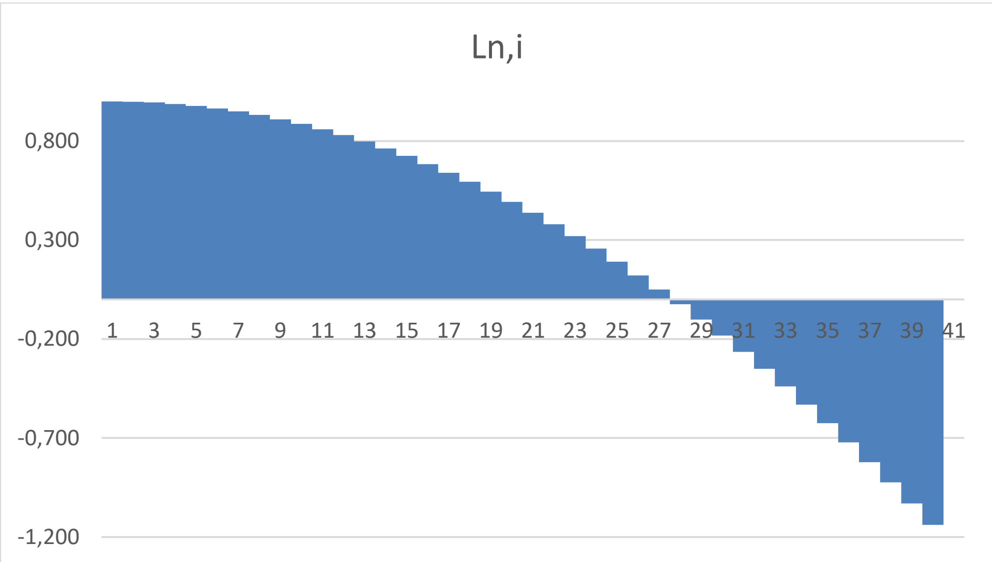
n=	10	int.			n=	40	int.			n=	100	int.		
h=	0,15	osztásköz			h=	0,0375	osztásköz			h=	0,015	osztásköz		
x	f(x)	$L_{n,i}$	$R_{n,i}$	$SM_{n,i}$	x	f(x)	$L_{n,i}$	$R_{n,i}$	$SM_{n,i}$	x	f(x)	$L_{n,i}$	$R_{n,i}$	$SM_{n,i}$
0,000	1,000	1,000			0,000	1,000	1,000			0,000	1,000	1,000		
0,150	0,978	0,978	0,978	0,989	0,038	0,999	0,999	0,999	0,999	0,015	1,000	1,000	1,000	1,000
0,300	0,910	0,910	0,910	0,944	0,075	0,994	0,994	0,994	0,996	0,030	0,999	0,999	0,999	0,999
0,450	0,798	0,798	0,798	0,854	0,113	0,987	0,987	0,987	0,991	0,045	0,998	0,998	0,998	0,999
0,600	0,640	0,640	0,640	0,719	0,150	0,978	0,978	0,978	0,982	0,060	0,996	0,996	0,996	0,997
0,750	0,438	0,438	0,438	0,539	0,188	0,965	0,965	0,965	0,971	0,075	0,994	0,994	0,994	0,995
0,900	0,190	0,190	0,190	0,314	0,225	0,949	0,949	0,949	0,957	0,090	0,992	0,992	0,992	0,993
1,050	-0,103	-0,103	-0,103	0,044	0,263	0,931	0,931	0,931	0,940	0,105	0,989	0,989	0,989	0,990
1,200	-0,440	-0,440	-0,440	-0,271	0,300	0,910	0,910	0,910	0,921	0,120	0,986	0,986	0,986	0,987
1,350	-0,823	-0,823	-0,823	-0,631	0,338	0,886	0,886	0,886	0,898	0,135	0,982	0,982	0,982	0,984
1,500	-1,250		-1,250	-1,036	0,375	0,859	0,859	0,859	0,873	0,150	0,978	0,978	0,978	0,980
					0,413	0,830	0,830	0,830	0,845	0,165	0,973	0,973	0,973	0,975
	$h^*\sum f_i$	0,538	0,201	0,369	0,450	0,798	0,798	0,798	0,814	0,180	0,968	0,968	0,968	0,970

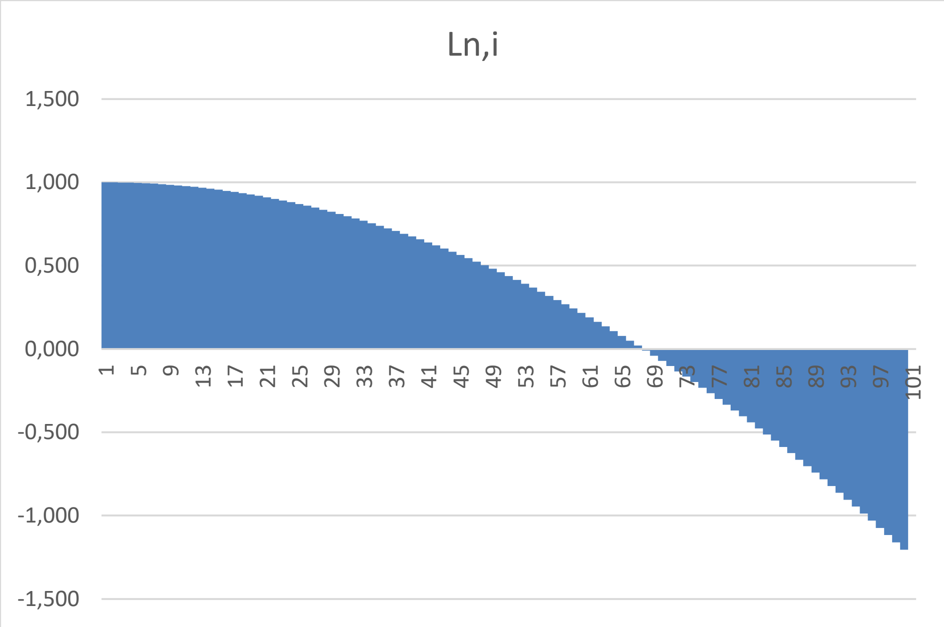
Összesítve a 3 különböző felosztás eredményeit, a következő értékeket kell kapnunk:

31	Összegzés:		$L_{n,i}$	$R_{n,i}$	$SM_{n,i}$
32		**h*∑f$_i$**	0,538	0,201	0,369
33		**h*∑f$_i$**	0,417	0,332	0,375
34		**h*∑f$_i$**	0,392	0,358	0,375

Ha csak a balkézmódszert ábrázolnánk, a következőképpen közelíti a 3 különböző felosztás a görbénket:







Számoljuk ki a tényleges függvény értékét, illetve a relatív hibákat, ha:

$$\int_0^{1,5} f(x)dx = \int_0^{1,5} (1-x^2)dx = \left(x - \frac{1}{3}x^3\right)\Big|_0^{1,5} = 1,5 - \frac{1,5^3}{3} = 0,375$$

Az egyes módszerek abszolút hibái pedig:

36		**Korrekt érték:**	0,375		
37					
38	Abszolút hibák:		$L_{n,i}$	$R_{n,i}$	$SM_{n,i}$
39		**ε**	0,163	0,174	0,006
40		**ε**	0,042	0,043	0,000
41		**ε**	0,017	0,017	0,000

A matematikai levezetés nélkül az összetett téglalap szabály becsült hibája balkéz (és jobbkéz) szabály esetén:

$$E = \left[\frac{1}{2}(b-a)\bar{f}'\right]h = O(h) \tag{5.9.}$$

ahol $\bar{f}'$ a függvény átlagos hibáját jelöli, melyet Taylor-sorba fejtéssel írhatunk fel, majd becsülhetjük meg értékét a teljes intervallumra.

Az összetett téglalap formula középpont szabály esetén a következő hibával rendelkezik:

$$E = \left[\frac{1}{24}(b-a)\bar{f}''\right]h^2 = O(h^2) \tag{5.10.}$$

ahol $\bar{f}''$ a becsült értéke f''-nek a teljes $[a,b]$ intervallum felett.

5.1.2 Trapéz szabály

Ahogy láttuk, a téglalap szabályok – a függvény változásának függvényében vagy alá- vagy fölé becsülik az integrál értéket. Ebből kifolyólag ésszerűnek tűnik, hogy a közelítés során vegyük a balkéz- és jobbkéz formulák számtani átlagát, ami jobban megfontolva nem más, mint az intervallum 4 pontja által meghatározott trapéz. Ezzel a két függvényérték - $f(a)$ és $f(b)$ – között egy lineáris vonalat; elsőrendű polinomot hoztunk létre. Az ez alatti terület pedig a trapéz területéből kiindulva elég egyszerűen kiadódik. Ha csak a balkéz- és jobbkéz szabályok átlagából indulunk ki:

$$T_n = \frac{L_n + R_n}{2} \tag{5.11.}$$

Ebből a geometriai interpretáció:

$$A = f(a) + \frac{f(b) - f(a)}{b-a}(x-a) \tag{5.12.}$$

Ezzel ismét abba a hibába esünk, mint az előbb, vagyis globális megoldást próbálunk keresni vele, holott érezhetjük, hogy ez a képlet önmagában még nem fogja elhozni a tejjel-mézzel folyó Kánaánt (lásd 14. ábra bal oldal). Ismét inkább érdemes ugyanezt többször megismételni az $[a,b]$ intervallumon, vagyis felosztva azt az egyes részintervallumok területeit írjuk fel (lásd 14. ábra jobb oldal):

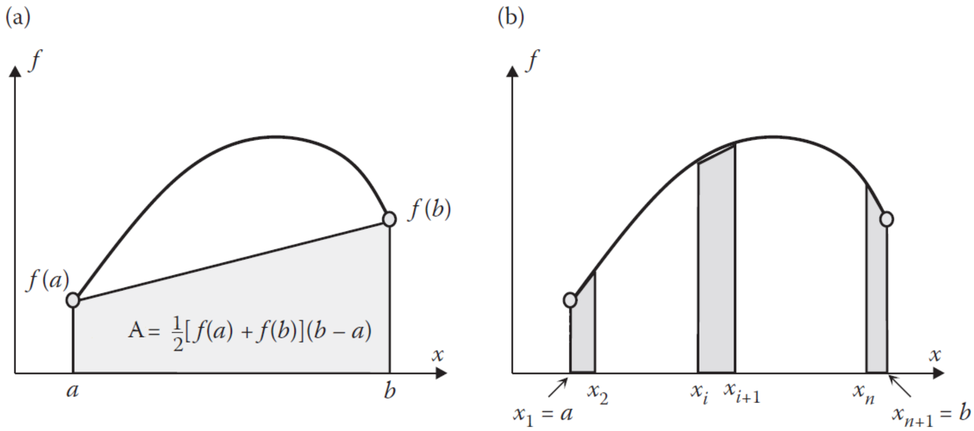
$$A_i = \frac{f(x_{i-1}) + f(x_i)}{2}\Delta x_i \tag{5.13.}$$

Ezzel tulajdonképpen fel is írhatjuk az összetett trapéz szabály formuláját:

$$T_n = \sum_{i=1}^{n} \frac{f(x_{i-1}) + f(x_i)}{2}\Delta x_i \tag{5.14.}$$

Ha a felosztást egyenlő osztásközökkel tesszük meg, észben tartva, hogy $x_{i+1} - x_i = h\ (i = 1,2,\dots,n)$, akkor a végképlet a következő lesz:

$$\begin{aligned} T_n &= \frac{h}{2}\sum_{i=1}^{n}[f(x_i) + f(x_{i+1})] = \\ &= \frac{h}{2}[f(a) + 2f(x_2) + 2f(x_3) + \cdots + 2f(x_n) + f(b)] \end{aligned} \tag{5.15.}$$



14. ábra: Az egyszerű (balra) és összetett (jobbra) trapéz szabályok

Az összetett trapéz formula a következő hibával rendelkezik:

$$E = \left[-\frac{1}{12}(b-a)\bar{f}''\right]h^2 = O(h^2) \qquad (5.16.)$$

ahol $\bar{f}''$ a becsült értéke f''-nek a teljes $[a, b]$ intervallum felett. Látható, hogy értéke az összetett középpont módszerével hasonló értékű, viszont a bal- és jobbkéz módszerekénél ($O(h)$) jelentősen jobb.

5.2. Példa

Feladat:

Folytassuk az előző táblázatot, és egészítsük ki a trapéz módszerrel is! (plusz oszlopokat beszúrva)

Megoldás:

Az F oszlopban az első elem szintén kimarad, majd F17-től indul a súlyozott függvényértékek számítása:

=(B17+B16)/2*(A17-A16)

Az összegzésnél viszont jó les figyelni, mert itt már nem kell bevennünk az osztásközt a buliba, elegendő csak összegeznünk az F oszlop elemeit.

A feladat ugyanez N és V oszlopokban is:

Numerikus integrálás - balkéz-, jobbkéz érték alapján

Vizsgáljuk meg az "f(x) = 1-x^2" függvény alatti területet a= 0 és b = 1,5 között 20 részintervallumra felosztva!

Adott:

a=	0,000
b=	1,500
n=	40,000
h=(b-a)/n=	0,038

Intervallumok számának megadása

n=	10	int.
h=	0,15	osztásköz

x	f(x)	$L_{n,i}$	$R_{n,i}$	$SM_{n,i}$	$T_{n,i}$
0,000	1,000	1,000			
0,150	0,978	0,978	0,978	0,989	0,148
0,300	0,910	0,910	0,910	0,944	0,142
0,450	0,798	0,798	0,798	0,854	0,128
0,600	0,640	0,640	0,640	0,719	0,108
0,750	0,438	0,438	0,438	0,539	0,081
0,900	0,190	0,190	0,190	0,314	0,047
1,050	-0,103	-0,103	-0,103	0,044	0,007
1,200	-0,440	-0,440	-0,440	-0,271	-0,041
1,350	-0,823	-0,823	-0,823	-0,631	-0,095
1,500	-1,250		-1,250	-1,036	-0,155

n=	40	int.
h=	0,0375	osztásköz

x	f(x)	$L_{n,i}$	$R_{n,i}$	$SM_{n,i}$	$T_{n,i}$
0,000	1,000	1,000			
0,038	0,999	0,999	0,999	0,999	0,037
0,075	0,994	0,994	0,994	0,996	0,037
0,113	0,987	0,987	0,987	0,991	0,037
0,150	0,978	0,978	0,978	0,982	0,037
0,188	0,965	0,965	0,965	0,971	0,036
0,225	0,949	0,949	0,949	0,957	0,036
0,263	0,931	0,931	0,931	0,940	0,035
0,300	0,910	0,910	0,910	0,921	0,035
0,338	0,886	0,886	0,886	0,898	0,034
0,375	0,859	0,859	0,859	0,873	0,033
0,413	0,830	0,830	0,830	0,845	0,032

n=	100	int.
h=	0,015	osztásköz

x	f(x)	$L_{n,i}$	$R_{n,i}$	$SM_{n,i}$	$T_{n,i}$
0,000	1,000	1,000			
0,015	1,000	1,000	1,000	1,000	0,015
0,030	0,999	0,999	0,999	0,999	0,015
0,045	0,998	0,998	0,998	0,999	0,015
0,060	0,996	0,996	0,996	0,997	0,015
0,075	0,994	0,994	0,994	0,995	0,015
0,090	0,992	0,992	0,992	0,993	0,015
0,105	0,989	0,989	0,989	0,990	0,015
0,120	0,986	0,986	0,986	0,987	0,015
0,135	0,982	0,982	0,982	0,984	0,015
0,150	0,978	0,978	0,978	0,980	0,015
0,165	0,973	0,973	0,973	0,975	0,015

Kiegészítve az előző összegző táblázatainkat, a következőképpen alakulnak az integrál értékek:

31	Összegzés:		$L_{n,i}$	$R_{n,i}$	$SM_{n,i}$	$T_{n,i}$
32		**h*∑f$_i$**	0,538	0,201	0,369	0,369
33		**h*∑f$_i$**	0,417	0,332	0,375	0,375
34		**h*∑f$_i$**	0,392	0,358	0,375	0,375
35						
36		**Korrekt érték:**		0,375		
37						
38	Abszolút hibák:		$L_{n,i}$	$R_{n,i}$	$SM_{n,i}$	$T_{n,i}$
39		**ε**	0,163	0,174	0,006	0,006
40		**ε**	0,042	0,043	0,000	0,000
41		**ε**	0,017	0,017	0,000	0,000

5.2 Kvadratúraformulák

A trapéz szabállyal eljutottunk az első fokú polinomokkal való munkáig, így a következő lépés a magasabb rendű polinomok használata lesz. Természetesen ennek következtében várhatóan az eredmény várható pontossága is magasabb lesz. Ezek a módszerek az ún. 1/3-os és 3/8-os Simpson szabályok, melyekkel másod- és harmadrendű polinomokat használunk fel a bemenő pontok, vagyis alappontok közelítésére. Tesszük mindezt továbbra is azért, hogy egy pontos, de mégis gyors módszerhez jussunk.

A Simpson szabályok felírása előtt viszont hazafias kötelességünk leszögezni, hogy innentől fogva kvadratúraformulákról beszélünk. Egy f függvény határozott integrálja az $[a, b]$ intervallumon közelíthető egy kvadratúra-formulával, ha:

- kiszámítjuk az alappontokat $(x_1, x_2, \dots, x_n)$, melyekre $a = x_1 < x_2 < \cdots < x_n = b$,
- meghatározunk minden x_i alapponthoz egy súlyt, w_i-t,
- A kvadratúraformula ekkor:

$$Q_n(f) = \sum_{i=1}^{n} w_i f(x_i) \tag{5.17.}$$

vagyis az alappontokon vett súlyozott függvényértékek összege.

Egy kvadratúra formula általában megkapható a következő formában:

- meghatározzuk az alappontokat $(x_1 \dots x_n)$

- a kvadratúra formula értékét az $(x_i, f(x_i))$ pontokra illesztett Lagrange-interpolációs polinom $[a, b]$-n vett integrálja legyen.

Ha ezt így adjuk meg, akkor interpolációs kvadratúra formuláról beszélünk. Tehát mivel a Lagrange interpolációs polinom felírható a következő alakban:

$$\sum_{i=1}^{n} f(x_i)L_i(x) \tag{5.18.}$$

ahol az $L_i(x)$ az adott x-hez tartozó i-edik Lagrange alappolinom. Ennek az integrálja pedig:

$$\int_a^b \sum_{i=1}^{n} f(x_i)L_i(x)dx = \sum_{i=1}^{n}\left(f(x_i)\int_a^b L_i(x)dx\right) \tag{5.19.}$$

vagyis egy interpolációs kvadratúra formula mindig kvadratúra formula lesz, ahol a megválasztott súly $w_i = \int_a^b L_i(x)dx$.

Newton-Cotes formulák

Gondolom evidens, hogy kvadratúra formulák használatakor is alapkérdés, hogy milyen alappontokat használunk, mivel Lagrange interpoláció esetén az alappontok esetében a súlypontok egyértelműen előállnak.

Ha a vizsgálandó $[a, b]$ intervallumot egyenlő osztásközzel osztjuk fel (ekvidisztánsan), a végpontjai pedig az alappontok, akkor az ezekre felírt interpolációs kvadratúra-formulát Newton-Cotes formuláknak nevezzük.

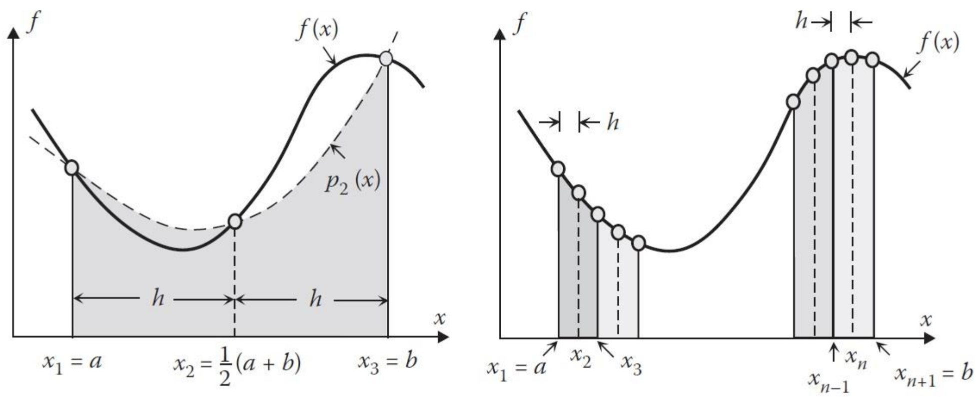
Összetett kvadratúra-szabályok

A polinommal történő integrálás pontossága nem feltétlenül javul az alappontok számának növelésével (az esetek jelentős részében igen, de nem mindig). Akárcsak az interpolációk esetében, egy bizonyos alappont-szám felett a közelítő polinom már elég zabolátlanul tud viselkedni. Ezen jelenség kiküszöbölése érdekében használatos a kiindulási $[a, b]$ intervallum felosztása n részre, melyekre külön-külön kvadratúra-formulákat írunk fel (pl. trapéz, vagy Simpson szabályt). Az ezen módszer szerint előállt kvadratúra formulát összetett kvadratúra-szabályoknak is nevezzük.

Meg kell jegyezni, hogy ezen témakört sokkal részletesebben is lehetne boncolgatni, viszont ezen jegyzet elsődleges célja, hogy a mérnöki gyakorlatban felhasználható módszerekre fókuszáljunk.

5.2.1 1/3-os Simpson szabály

Az eddigi módszerekkel szemben az 1/3-os Simpson szabály másodfokú polinomokkal (nem véletlen a többes szám!) közelíti az alappontokat. Ehhez 3 pontra van szükségünk egyszerre, melyek a következő módon követik egymást: $x_1 = a$, $x_2 = (a+b)/2$ és $x_3 = b$, ahogy ezt a 15. ábra bal oldala is szemlélteti.



15. ábra: Egyszerű 1/3-os Simpson szabály (balra), összetett 1/3-os Simpson szabály (jobbra)

A legegyszerűbb megoldás a másodfokú görbe illesztésére a másodfokú Lagrange interpolációs polinom alkalmazása, mely felírható, mint:

$$p_2(x) = \frac{(x-x_2)(x-x_3)}{(x_1-x_2)(x_1-x_3)} f(x_1) + \frac{(x-x_1)(x-x_3)}{(x_2-x_1)(x_2-x_3)} f(x_2) + \\ + \frac{(x-x_1)(x-x_2)}{(x_3-x_1)(x_3-x_2)} f(x_3) \tag{5.20.}$$

Ezzel az integrálandó formula a következő lesz:

$$\int_a^b f(x)dx = \int_a^b p_2(x)dx \tag{5.21.}$$

Ha behelyettesítjük $p_2(x)$ polinomot $x_1 = a, x_2 = \frac{a+b}{2}, x_3 = b$ értékekkel, és elvégezzük az integrálást a-tól b-ig, akkor a következő képletet kapjuk, mely egyben az egyszerű 1/3-os Simpson szabály képlete is:

$$\int_a^b f(x)dx \cong \frac{h}{3}[f(x_1) + 4f(x_2) + f(x_3)] \tag{5.22.}$$

ahol $h = \frac{b-a}{2}$. Ez a módszer hihetetlen kreativitásról tanúbizonyságot téve az $1/3$-os szorzóról kapta a nevét.

Az összetett $1/3$-os Simpson szabály esetében az $[a, b]$ intervallum felosztásra kerül n darab részintervallumra, ehhez $n + 1$ (!) pontot használnunk fel, így $a = x_1, x_2, \dots, x_n, x_{n+1} = b$. Habár megoldható, hogy az intervallumok szélessége nem ugyanakkora, az ekvidisztáns felosztás ($h = \frac{(b-a)}{n}$) sokkal egyszerűbbé teszi életünket. Mivel egy részintervallumra való parabola illesztéshez három pont szükséges, így különösen fontos szabály, hogy az 1/3-os Simpson szabály akkor alkalmazható, ha a részintervallumok száma páros! Ezzel az összetett 1/3-os Simpson szabály képlete a következőképp néz ki:

$$\begin{aligned}\int_a^b f(x)dx \cong \frac{h}{3}[f(x_1) + 4f(x_2) + f(x_3)] + \frac{h}{3}[f(x_3) + 4f(x_4) + f(x_5)] + \cdots \\ + \frac{h}{3}[f(x_{n-1}) + 4f(x_n) + f(x_{n+1})]\end{aligned} \tag{5.23.}$$

Látható, hogy az összegzés során a páros indexű függvény értékeket $(x_2, x_4, \dots, x_n)$ négyszer adjuk össze, míg a páratlan indexű függvény értékeket $(x_3, x_5, \dots, x_{n-1})$ csak kétszer míg az első és utolsó függvény értéket csak egyszer-egyszer. Így az összetett Simpson szabály formulája összevont alakban:

$$\int_a^b f(x)dx \cong \frac{h}{3}\left\{f(x_1) + 4\sum_{i=2,4,6\dots}^{n} f(x_i) + 2\sum_{j=3,5,7\dots}^{n-1} f(x_j) + f(x_{n+1})\right\} \tag{5.24.}$$

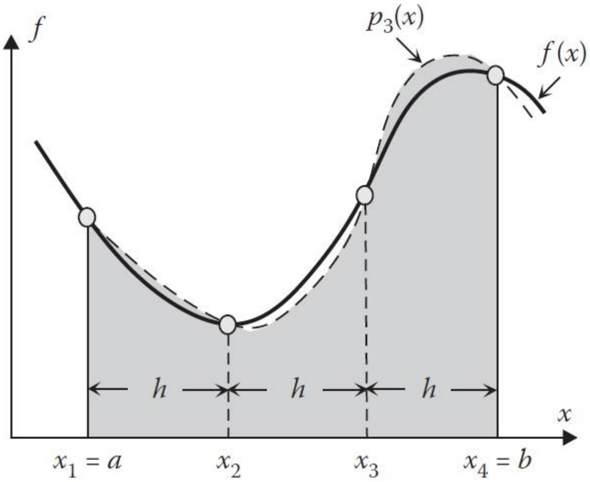
Végezetül rántsuk le a leplet, hogy miért is szeretjük annyira ezt a módszert! Az összetett 1/3-os Simpson módszer hibája:

$$E = \left[-\frac{1}{180}(b-a)\bar{f}^{(4)}\right]h^4 = O(h^4) \tag{5.25.}$$

ahol $\bar{f}^{(4)}$ a becsült értéke $f^{(4)}$-nek a teljes $[a, b]$ intervallum felett. Vagyis az összetett trapéz módszer hibájához képest ($O(h^2)$) ez a Simpson módszer két nagyságrenddel pontosabb ($O(h^4)$) megoldást képes szolgáltatni.

5.2.2 3/8-os Simpson szabály

Az 1/3-os Simpson szabály továbbfejlesztése, amikor nem másod-, hanem harmadrendű polinomot alkalmazunk az $f(x)$ függvény értékének közelítése érdekében. Ehhez 4 pontra van szükségünk, melyek között – hasonlóan az eddig megfontolásokhoz – ugyanakkora távolság van, így: $x_1 = a, x_2 = \frac{2a+b}{3}, x_3 = \frac{a+2b}{3}, x_4 = b$, ahol $h = \frac{b-a}{3}$, ahogy ezt a 16. ábra is mutatja.



16. ábra: 3/8-os Simpson szabály

A harmadfokú Lagrange polinom a 4 alappontra a következőképpen írható fel:

$$p_3(x) = \frac{(x-x_2)(x-x_3)(x-x_4)}{(x_1-x_2)(x_1-x_3)(x_1-x_4)}f(x_1) + \frac{(x-x_1)(x-x_3)(x-x_4)}{(x_2-x_1)(x_2-x_3)(x_2-x_4)}f(x_2) + \\ + \frac{(x-x_1)(x-x_2)(x-x_4)}{(x_3-x_1)(x_3-x_2)(x_3-x_4)}f(x_3) + \frac{(x-x_1)(x-x_2)(x-x_3)}{(x_4-x_1)(x_4-x_2)(x_4-x_3)}f(x_4) \qquad (5.26.)$$

Ha ezt a polinomot helyettesítjük vissza az integrálandó függvény helyére:

$$\int_a^b f(x)dx \cong \int_a^b p_3(x)dx \qquad (5.27.)$$

Ha a $p_3(x)$ polinomban x_1, x_2, x_3, x_4 értékeket behelyettesítjük a és b paraméterekkel, akkor a 3/8-os Simpson szabály a következőképpen írható fel:

$$\int_a^b f(x)dx \cong \frac{3h}{8}[f(x_1) + 3f(x_2) + 3f(x_3) + f(x_4)] \qquad (5.28.)$$

Természetesen az elnevezés most is a 3/8-ados szorzóból ered.

Persze most sem állhatunk meg egy globális módszernél, a 3/8-os Simpson szabálynak szintén van összetett verziója, ahol a vizsgált $[a, b]$ tartományt $n + 1$ darab alappontra osztjuk fel: $a = x_1, x_2, \ldots, x_n, x_{n+1} = b$. Célszerű továbbra is az ekvidisztáns felosztáshoz ragaszkodni. Mivel a 3/8-os Simpson szabály harmadfokú polinomjának felírásához 4 pont szükséges, nagyon fontos szabály, hogy az $[a, b]$ intervallum felosztásánál 3-mal osztható számú részintervallumot kell létrehozni! Ekkor a felosztás után a következőképpen összegezhetjük a részintervallumok területeit:

$$\begin{aligned}\int_a^b f(x)dx \cong \frac{3h}{8}[f(x_1) + 3f(x_2) + 3f(x_3) + f(x_4)] \\ + \frac{3h}{8}[f(x_4) + 3f(x_5) + 3f(x_6) + f(x_7)] + \cdots \\ + \frac{3h}{8}[f(x_{n-2}) + 3f(x_{n-1}) + 3f(x_n) + f(x_{n+1})]\end{aligned} \tag{5.29.}$$

Ha elkezdjük összevonogatni az egyes tagokat, akkor láthatjuk, hogy a $2., 5., 8., \ldots$ és $3., 6., 9., \ldots$ indexű elemek 3-as szorzóval összegezhetők, míg a $4., 7., 10., \ldots$ indexű elemek 2-es szorzóval, vagyis összevonva így írható fel:

$$\int_a^b f(x)dx \cong \frac{3h}{8}\left\{f(x_1) + 3\sum_{i=2,5,8}^{n-1}[f(x_i) + f(x_{i+1})] + 2\sum_{j=4,7,10}^{n-2} f(x_j) + f(x_{n+1})\right\} \tag{5.30.}$$

Ezzel el is jutottunk az összetett 3/8-os Simpson szabály képletéhez. Ezen módszer pontossága a következőképpen írható fel:

$$E = \left[-\frac{1}{80}(b-a)\overline{f^{(4)}}\right]h^4 = O(h^4) \tag{5.31.}$$

ahol $\overline{f^{(4)}}$ a becsült értéke $f^{(4)}$-nek a teljes $[a, b]$ intervallum felett. Vagyis az összetett trapéz módszer hibájához képest ($O(h^2)$) ez a Simpson módszer két nagyságrenddel pontosabb ($O(h^4)$) megoldást képes szolgáltatni, ugyanakkor pontossága összemérhető az összetett, 1/3-os Simpson szabállyal.

Habár léteznek más megközelítések is, például ha az integrálandó függvény adott, a Romberg és Gauss kvadratúraformulák igen jó szolgáltatást adnak, az összetett Simpson módszerek a legtöbb mérnöki problémára gyors és kellően pontos eredményt képesek szolgáltatni, így alkalmazásuk igen elterjedt.

5.3. Példa

Feladat:

Ha már ennyire bejött eddig, akkor folytassuk tovább az eddig megkezdett feladatot, és vizsgáljuk meg 1/3-os Simpson módszerrel is 10 és 40 részintervallumra felosztva [a,b]-t!

Megoldás:

Habár kiférne egy táblázatba, hogy ne folyjon össze, nyissunk egy új munkalapot, és hozzuk létre az eddigi séma alapján a táblázatunkat:

Numerikus integrálás Simpson szabály segítségével

Vizsgáljuk meg az "f(x) = 1-x²" függvény alatti területet a= 0 és b = 1,5 között 20 részintervallumra felosztva!

Adott:

a=	0,000
b=	1,500
n=	40,000
h=(b-a)/n=	0,038

Intervallumok számának megadása

n=	10,000	int.
h=	0,150	osztásköz

Sorszám	x	f(x)	$S_{n,i,1/3}$	Szorzó	Szorzó	c*f(x$_i$)
1	0,000	1,000	1,000	1,000		1,000
2	0,150	0,978	0,978	4,000		3,910
3	0,300	0,910	0,910		2,000	1,820
4	0,450	0,798	0,798	4,000		3,190
5	0,600	0,640	0,640		2,000	1,280
6	0,750	0,438	0,438	4,000		1,750
7	0,900	0,190	0,190		2,000	0,380
8	1,050	-0,103	-0,103	4,000		-0,410
9	1,200	-0,440	-0,440		2,000	-0,880
10	1,350	-0,823	-0,823	4,000		-3,290
11	1,500	-1,250	-1,250	1,000		-1,250
					$\sum S_{n,1/3}$	**0,375**

Intervallumok számának megadása

n=	40,000	int.
h=	0,038	osztásköz

Sorszám	x	f(x)	$S_{n,i,1/3}$	Szorzó	Szorzó	c*f(x$_i$)
1	0,000	1,000	1,000	1,000		1,000
2	0,038	0,999	0,999	4,000		3,994
3	0,075	0,994	0,994		2,000	1,989
4	0,113	0,987	0,987	4,000		3,949
5	0,150	0,978	0,978		2,000	1,955
6	0,188	0,965	0,965	4,000		3,859
7	0,225	0,949	0,949		2,000	1,899
8	0,263	0,931	0,931	4,000		3,724
9	0,300	0,910	0,910		2,000	1,820
10	0,338	0,886	0,886	4,000		3,544
11	0,375	0,859	0,859		2,000	1,719
12	0,413	0,830	0,830	4,000		3,319
13	0,450	0,798	0,798		2,000	1,595

Az eddigiekhez képesti újdonság a D oszlop mellett (ahol már minden f(x) értéket beveszünk a buliba) az E és F oszlopban lévő szorzók meghatározása a képlet szerint. Ehhez figyelni kell az osztásközre, hiszen ahogy már mondtuk, páros számú intervallumra lesz szükség, így a sorok számát meg kell okosítani kicsit.

A G oszlop elemei a következőképpen állnak fel:

=D16*HA(E16<>0;E16;F16)

Ezzel biztosíthatjuk, hogy azzal a szorzóval fogja elvégezni a műveletet, amire tényleg szükségünk van.

Végül az összegzésnél (G28) pedig:

=($B$12/3)*SZUM(G16:G26)

A pontosság pedig maga a tökély már a 10-es felosztásnál is:

Összegzés:		$L_{n,i}$	$R_{n,i}$	$SM_{n,i}$	$T_{n,i}$	$S_{n,i,1/3}$
n=10	**h*∑f$_i$**	0,538	0,201	0,369	0,369	0,375
n=40	**h*∑f$_i$**	0,417	0,332	0,375	0,375	0,375
n=100	**h*∑f$_i$**	0,392	0,358	0,375	0,375	0,375
	Korrekt érték:		0,375			
Abszolút hibák:		$L_{n,i}$	$R_{n,i}$	$SM_{n,i}$	$T_{n,i}$	$S_{n,i,1/3}$
n=10	**ε**	0,163	0,174	0,006	0,006	0,000
n=40	**ε**	0,042	0,043	0,000	0,000	0,000
n=100	**ε**	0,017	0,017	0,000	0,000	0,000

5.4. Példa

Feladat:

Na, meglepetés ha azt mondom, hogy ugyanez, mint eddig? Csak most irány a 3/8-os Simpson módszer!

Megoldás:

Mivel a módszer ugyanúgy működik, mint az 1/3 Simpson, így csak az értékeket mutatom meg:

	A	B	C	D	E	F	G	H	I
1	Numerikus integrálás Simpson szabály segítségével								
2									
3	Vizsgáljuk meg az "f(x) = 1-x^2" függvény alatti területet a= 0 és b = 1,5 között 20 részintervallumra felosztva!								
4	Adott:								
5	a=	0,000							
6	b=	1,500							
7	n=	40,000							
8	h=(b-a)/n=	0,038							
9									
10	Intervallumok számának megadása								
11	n=	9,000	int.						
12	h=	0,167	osztásköz						
13									
14									
15	Sorszám	x	f(x)	$S_{n,i,1/3}$	Szorzó	Szorzó	Szorzó	c*f(x$_i$)	
16	0	0,000	1,000	1,000	1,000			1,000	
17	1	0,167	0,972	0,972	3,000			2,917	
18	2	0,333	0,889	0,889		3,000		2,667	
19	3	0,500	0,750	0,750			2,000	1,500	
20	4	0,667	0,556	0,556	3,000			1,667	
21	5	0,833	0,306	0,306		3,000		0,917	
22	6	1,000	0,000	0,000			2,000	0,000	
23	7	1,167	-0,361	-0,361	3,000			-1,083	
24	8	1,333	-0,778	-0,778		3,000		-2,333	
25	9	1,500	-1,250	-1,250	1,000			-1,250	
26									
27							∑ $S_{n,3/8}$	**0,375**	

	J	K	L	M	N	O	P	Q
10	Intervallumok számának megadása							
11	n=	39,000	int.					
12	h=	0,038	osztásköz					
13								
14								
15	Sorszám	x	f(x)	$S_{n,i,1/3}$	Szorzó	Szorzó	Szorzó	c*f(x$_i$)
16	0	0,000	1,000	1,000	1,000			1,000
17	1	0,038	0,999	0,999	3,000			2,996
18	2	0,077	0,994	0,994		3,000		2,982
19	3	0,115	0,987	0,987			2,000	1,973
20	4	0,154	0,976	0,976	3,000			2,929
21	5	0,192	0,963	0,963		3,000		2,889
22	6	0,231	0,947	0,947			2,000	1,893
23	7	0,269	0,928	0,928	3,000			2,783
24	8	0,308	0,905	0,905		3,000		2,716
25	9	0,346	0,880	0,880			2,000	1,760
26	10	0,385	0,852	0,852	3,000			2,556
27	11	0,423	0,821	0,821		3,000		2,463

5.5. Példa

Feladat:

Hozzunk létre egy olyan táblázatot, mely alkalmas arra, hogy egy megadott függvényt integráljon a felsorolt módszerekkel egy makró végrehajtásával, melyet egy gomb segítségével tudunk indítani. Írjuk ki az összes eredményt, a vizsgált függvényt pedig a makróban definiáljuk! A futtatáshoz a függvényt, az alsó és felső határt illetve a részintervallumok számát kell megadnunk, továbbá a primitív függvényt, hogy számítani lehessen a relatív hibáját az egyes algoritmusoknak.

Megoldás:

A végleges táblázat a következőképpen néz ki:

	A	B	C	D	E	F	G	H
1	**Numerikus integrálás VBA használatával**							
2								
3	Ezen az excel táblázatban a különböző numerikus integrálási módszerek megoldásait vetjük össze.							
4								
5	**Bemenet**							
6	Példa függvény:		**f(x) = X^3 + 2X - 0.15**					
7		Alsó határ =	0,00					
8		Felső határ =	2,4		Ezeket az adatokat kell megadni.			
9	Részintervallumok száma=		12					
10		h=	0,2		számított lépésköz			
11								
12	**Számítás**							
13	A "Függvény változtatása" gomb elvisz a makróban oda, ahol a függvényt definiáljuk.							
14	Az "Integrálás!" gombbal indítjuk el a makrót.							
15								
16		Függvény változtatása		Integrálás!				
17								
18			**Eredmények**					
19		Függvény alatti területek		Relatív hibák [%]				
20								
21		Pontos érték	13,6944					
22		Balkéz Szabály	11,8896		13,179108			
23		Jobbkéz Szabály	15,6144		14,020332			
24		Trapéz módszer	13,752		0,4206113			
25		1/3 Simpson szabály	13,6944		8,166E-07			
26		3/8 Simpson szabály	13,6944		1,302E-06			

A megoldó program a következőképpen néz ki:

Module 1:

```
Dim x(1000000) As Single, y(1000000) As Single
Sub Integrate()
'5 különböző numerikus integrálási módszer
'x - független változó array
'y - függő változó array
'n - intervallumok száma
'h - lépésköz
'LB - alsó határ
'UB - felső határ

Dim n, i As Integer
Dim h As Double
'adatok beolvasása
lb = Cells(7, 3).Value
ub = Cells(8, 3).Value
n = Cells(9, 3).Value
h = (ub - lb) / n
x(0) = lb
y(0) = F_of_x(lb)

For i = 1 To n  'x és y array-ek létrehozása
   x(i) = x(0) + h * i
   y(i) = F_of_x(x(i))
Next i

Cells(21, 3).Value = (IF_of_x(ub) - IF_of_x(lb))
Cells(22, 3).Value = LR(y, h, n)
Cells(23, 3).Value = RR(y, h, n)
Cells(24, 3).Value = TR(y, h, n)
Cells(25, 3).Value = SR(y, h, n)
Cells(26, 3).Value = S38R(y, h, n)

End Sub

Function SR(y, h, n)

SR = 0  'inicializálás
```

```
For i = 2 To n Step 2
   SR = SR + (h / 3) * (y(i - 2) + 4 * y(i - 1) + y(i))
Next i
End Function

Function TR(y, h, n)

TR = 0 'inicializálás
For i = 1 To n
   TR = TR + (h / 2) * (y(i - 1) + y(i))
   Next i

End Function

Function LR(y, h, n)

LR = 0
For i = 0 To n - 1
   LR = LR + h * y(i)
   Next i

End Function

Function RR(y, h, n)

RR = 0
For i = 1 To n
   RR = RR + h * y(i)
   Next i

End Function

Function S38R(y, h, n)

S38R = 0  ' inicializálás
For i = 3 To n Step 3
   S38R = S38R + (3 * h / 8) * (y(i - 3) + 3 * y(i - 2) + 3 * y(i - 1) + y(i))
Next i
```

```
End Function

Function F_of_x(x)
'Ezt a függvényt fogjuk integrálni, melyet kézzel kell bevinni

F_of_x = x ^ 3 + 2 * x - 0.15

End Function

Function IF_of_x(x)

'Ez pedig a primitív függvény, melyet szintén kézzel viszünk be
IF_of_x = x ^ 4 / 4 + x ^ 2 - 0.15 * x

End Function
```

Module 2:

```
Sub Alter_funct()
' A függvény változtatásához irányít a makró vonatkozó részéhez
'Az elhelyezett gombbal lehet aktiválni

   Application.Goto Reference:="f_of_x"
   Windows("integralas.xlsm").Activate
End Sub

Sub Run_SR_Macro()
' Futtatjuk SR_Macro makrót.
' Az elhelyezett gombbal lehet aktiválni

   Application.Run "'integralas.xlsm'!integrate"
End Sub
```

6 Optimalizáció

Az optimalizáció általános értelmezésében olyan problémák és megoldások összessége, melyek a rendelkezésre álló erőforrások legjobb (optimális) elosztását teszi lehetővé. Maga az optimalizáció, mint problémakör, egy egész tantárgyat érdemelne, így jelen fejezet éppen hogy csak megkaparja a felszínt.

A hétköznapi optimalizációs problémák során lineáris modellek használhatók, melyek egyenletek és egyenlőtlenségek (kényszerek) alakjában írhatók fel. Az ilyen problémák többféleképpen is megoldhatók, de az egyik legelterjedtebb módszer – melyet a programozásnál is használnak – a simplex módszer.

Egy megfelelően kezelt modell akár több ezer kényszert is tud kezelni egyszerre. Ezt a módszert igen széles körben alkalmazzák: termelési tervek, szállítmányozás, raktározás, közlekedés-modellezés, energiatermelés-optimalizálás, de akár károsanyag-kibocsátás minimalizálásra is.

6.1 Alap megfontolások

Az **optimalizálás célja** mindig egy f függvény optimalizálása (maximalizálása vagy minimalizálása. Így f függvény lesz a célfüggvény.

A legtöbb optimalizációs probléma több változó által befolyásolt: $x_1, \dots, x_n$.

Ezeket **irányító változók**nak hívjuk, mert mi határozzuk meg őket, mi „irányítjuk".

Az optimalizálás-elmélet ezen $x_1, \dots, x_n$ paraméterek olyan meghatározását jelenti, mely az adott f függvény maximális vagy minimális legyen.

Viszont vannak olyan paraméterek, melyek nem választhatók meg szabadon, ezek **kényszer**ekként jelennek meg a modellben.

Már ismert fogalmak egy függvény maximuma, minimuma (összegezve extrémuma) egy teljes vagy egy rész-intervallumon.

Ezek alapján, ha f függvény differenciálható R intervallumon, és extrémuma van X_0 pontban, akkor a parciális deriváltaknak 0-nak kell lennie X_0 pontban. Ezt már elneveztük anno gradiensnek:

$$\nabla f(X_0) = 0 \qquad (6.1.)$$

Ez egy szükséges, de nem elégséges feltétel, gondoljunk csak a nyeregpontra (pl.: $y = x^3$)

Az előző problémák kiküszöbölése érdekében használnak iterációs módszereket, a fals eredmények kiküszöbölése érdekében. A legmeredekebb csökkenés / gradiens módszer ezen az elven nyugszik.

6.2 Legmeredekebb csökkenés módszere

A legmeredekebb csökkenés módszere (alias „method of steepest descent" - MSD) egy kereső irányos iteratív módszer. Lényege, hogy megtaláljuk a $f(x)$ függvény minimumát újra és újra, egy $g(t)$ függvény minimumának megkeresése segítségével, egy (t) változót felhasználva:

Tegyük fel, hogy f-nek minimuma van X_0-ban, és mi x pontban kezdjük el a vizsgálatot. Aztán megkeressük f-nek egy minimumát közel x-hez a $-\nabla f(x)$ irányban – ami a legmeredekebb iránya f-nek x körül:

$$z(t) = x - t\nabla f(x) \tag{6.2.}$$

Ahol minimuma van a következő függvénynek:

$$g(t) = f(z(t)) \tag{6.3.}$$

Ezután $z(t)$-t vesszük alapul X_0-ig.

6.1. Példa:

Feladat:

Keressük a következő függvény minimumát:

$$f(x) = x_1^2 + 3x_2^2$$

A kiindulási pont pedig legyen:

$$x_0 = (6; 3) = 6i + 3j$$

Megoldás:

1. Nyilvánvaló, hogy $f(x)$-nek minimuma van 0-ban. Legyen:

   $$\nabla f(x) = 2x_1 i + 6x_2 j$$

   Amiből:

   $$z(t) = x - t\nabla f(x) = (1-2t)x_1 i + (1-6t)x_2 j$$
   $$g(t) = f\big(z(t)\big) = (1-2t)^2 x_1^2 + 3(1-6t)^2 x_2^2$$

2. Így számíthatjuk a deriváltat:

   $$g'(t) = 2(1-2t)x_1^2(-2) + 6(1-6t)x_2^2(-6)$$

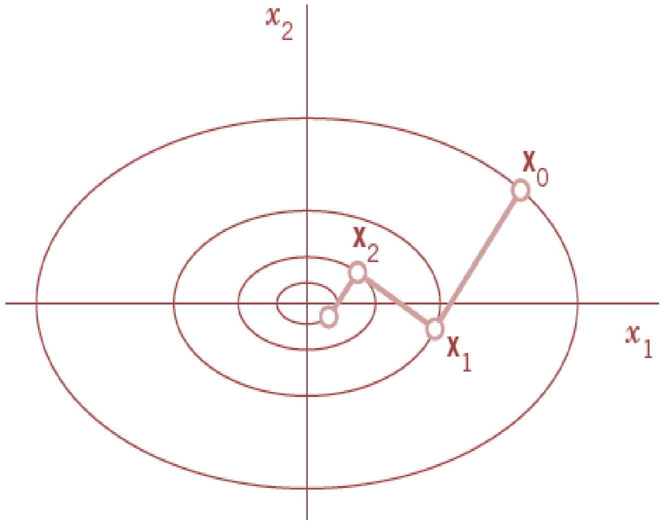
   Megadjuk $g'(t) = 0$-t, megoldjuk t-re:

   $$t = \frac{x_1^2 + 9x_2^2}{2x_1^2 + 54x_2^2}$$

Vagyis ha végig megyünk $x_0 = 6i + 3j$-től, a következő táblázatban foglalhatjuk össze az eredményeket:

N	x		t	1-2t	1-6t
0	6,000	3,000	0,210	0,581	-0,258
1	3,484	-0,774	0,310	0,381	-0,857
2	1,327	0,664	0,210	0,581	-0,258
3	0,771	-0,171	0,310	0,381	-0,857
4	0,294	0,147	0,210	0,581	-0,258
5	0,170	-0,038	0,310	0,381	-0,857
6	0,065	0,032			

Megjelenítve az algoritmus működését a következő módon találja meg a függvényünk extrémumát:



6.3 Lineáris optimalizáció

A lineáris optimalizáció során egy lineáris függvénnyel/függvényekkel leírható érték, mennyiség extrémumát akarjuk meghatározni. Amennyiben csak nem negatív megoldásokat keresünk, mellékfeltételként egyenlőtlenségeket kell definiálni. A feladat gyorsabb megértése érdekében értelmezzük egy példán keresztül!

6.2. Példa

Feladat:

Egy cég két fajta terméket gyárt: S-t és L-t. Az S termék 40€, az L 88€. A gyártó robotsorok időkorlátja a következő: M1 robotsor 2 perc alatt gyárt egy S-, és 8 perc alatt egy L terméket, M2 robotsor 5 perc alatt gyártja S-t, és 2 perc alatt L terméket. Határozzuk meg, hogy melyik gép mit gyártson óránként, ha:

$$z = f(x) = 40x_1 + 88x_2$$

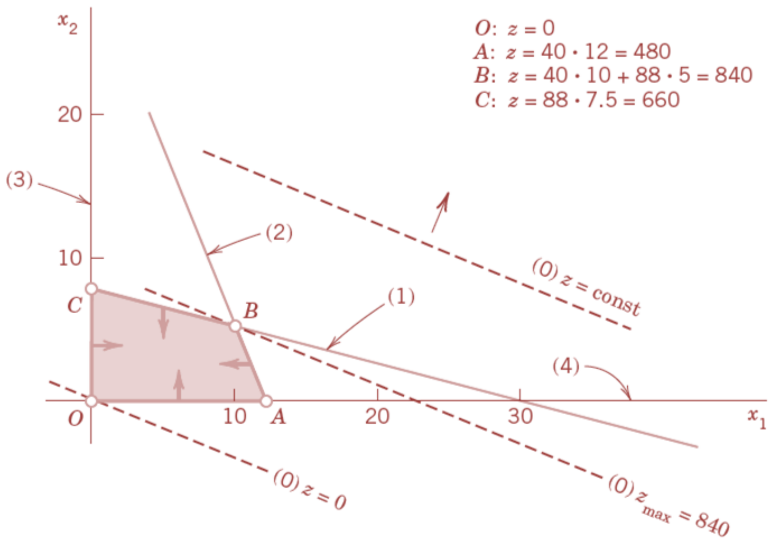
Megoldás:

Az biztos, hogy x_1 és x_2 nemnegatív. Vagyis:

0) $z = 40x_1 + 88x_2$
1) $2x_1 + 8x_2 \leq minimum\ idő\ M1 - en$
2) $5x_1 + 2x_2 \leq minimum\ idő\ M2 - n$
3) $x_1 \geq 0$
4) $x_2 \geq 0$

Mivel csak két változónk van, így a feladat megoldható grafikusan:

$$z_{max} = 40 * 10 + 88 * 5 = 840€$$



Ezeknél a feladatoknál nem megoldás, ha az egyik változót 0 értékre vesszük fel, mivel kritikus a régió, amely értékeket az irányító változók felvehetnek

Továbbá a grafikus megoldás csak két változó esetén működőképes, és ez csak a megoldandó feladatok kis százalékában alkalmazható.

6.4 Lineáris optimalizáció általános alakja

Visszatérve a kiindulási egyenlethez, a következőképpen indulhatunk el:

$$2x_1 + 8x_2 \leq 60$$

E kifejezésből következik, hogy a következő mennyiség nem negatív:

$$x_3 = 60 - 2x_1 - 8x_2$$

Vagyis, az eredeti egyenlet átírható a következő alakba:

$$2x_1 + 8x_2 + x_3 = 60$$

Ahol

$$x_3 \geq 0$$

Ezzel a művelettel becsempésztük x_3 nemnegatív segédváltozót, hogy segítségével az egyenlőtlenségeket egyenletekké konvertálhassuk. Az ilyen változót **tétlen változónak** hívjuk.

Egy általános lineáris optimalizációs probléma a következő általános alakba írható:

$$f = c_1x_1 + c_2x_2 + \cdots + c_nx_n \tag{6.4.}$$

Az egyes kényszerekre behelyettesítve:

$$\begin{gathered} a_{11}x_1 + \cdots + a_{1n}x_n = b_1 \\ a_{21}x_1 + \cdots + a_{2n}x_n = b_2 \\ \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \\ a_{m1}x_1 + \cdots + a_{mn}x_n = b_m \\ x_i \geq 0 \quad i = 1, \dots, n \end{gathered} \tag{6.5.}$$

Minden b_i elem nemnegatív (ha mégis az lenne, -1-gyel szorozni kell). x_i elemek tartalmazzák a tétlen változókat is. Az egyenletek lineárisan függetlenek. Ekkor ha értéket adunk $n - m$–ig változóknak, akkor a rendszer egyedülálló módon determinálja a többit. Vagyis:

$$x_1 \geq 0, \dots, x_n \geq 0 \tag{6.6.}$$

Ez a választás viszont nem teljesen szabad.

Egy n-szeres $(x_1, \dots, x_n)$-t, mely kielégíti a kényszereket, megvalósítható pontnak, **megvalósítható megoldásnak** tekintjük.

Egy megvalósítható megoldás pedig **optimális megoldás**, ha a célfüggvény (f) maximum lesz, az összes megvalósítható megoldás közül.

Általános megvalósítható megoldás alatt olyan megvalósítható megoldást értünk, amihez legalább n-m darab változó értéke 0 lesz $x_1, \dots, x_n$ közül. Vagyis az is kijelenthető, hogy egy lineáris optimalizációs probléma optimális megoldása egyben egy általános megvalósítható megoldás.

Az előző részből tovább gondolva a problémát legyen adott egy lineáris optimalizációs probléma:

Maximalizáljuk:

$$z = f(x) = c_1x_1 + \cdots + c_nx_n$$

Az alábbi kényszerekkel:

$$a_{11}x_1 + \cdots + a_{1n}x_n = b_1$$
$$a_{21}x_1 + \cdots + a_{2n}x_n = b_2$$
$$\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots$$
$$a_{m1}x_1 + \cdots + a_{mn}x_n = b_m$$
$$x_i \geq 0 \quad i = 1, \dots, n$$

Az optimális megoldás megtalálásához általános megoldásokat kell találni, viszont ezekből túl sok van, így valamilyen szisztematikus keresési módszer megtalálására van szükség. Ezt a módszert nevezzük **simplex** módszernek. A módszer lépésenként vizsgálja az általános megvalósítható megoldásokat, miközben a célfüggvény f folyton növeli értékét.

A kiindulási feladatunkat tovább gondolva:

$$z = 40x_1 + 88x_2$$
$$2x_1 + 8x_2 \leq 60$$
$$5x_1 + 2x_2 \leq 60$$
$$x_1 \geq 0$$
$$x_2 \geq 0$$

x_3 és x_4 tétlen változók bevezetésével az első két egyenlőtlenség átalakítható a probléma normál alakjává:

$$z - 40x_1 - 88x_2 = 0$$
$$2x_1 + 8x_2 + x_3 = 60$$
$$5x_1 + 2x_2 + x_4 = 60$$

Ahol:

$$x_1 \geq 0, \dots, x_4 \geq 0$$

Ez egy lineáris egyenletrendszer. Az optimális megoldás megtalálásához létre kell hozni az **induló táblázatot**.

$$T_0 = \begin{array}{c|c:cc:cc:c|} \multicolumn{1}{c}{} & z & x_1 & x_2 & x_3 & x_4 & \multicolumn{1}{c}{b} \\ \hline & 1 & -40 & -88 & 0 & 0 & 0 \\ \hdashline & 0 & 2 & 8 & 1 & 0 & 60 \\ & 0 & 5 & 2 & 0 & 1 & 60 \end{array}$$

Minden induló táblázat kétféle változó x_i –t tartalmaz. **Alapváltozók** alatt azokat az oszlopokat értjük, melyekben csak egy nemzérus elem van. Vagyis esetünkben x_3 és x_4 alapváltozók, x_1 és x_2 pedig nem alapváltozók.

- A felső sort a primál változók elemei adják
- Az alsó sorban a célfüggvény együtthatói helyezkednek el
- Jobb oldalon a készletvektor elemei vannak

Minden táblázat tartalmaz egy általános megvalósítható megoldást, melyeket a nem alapváltozók nullára állításával lehet kifejezni.

Vagyis alap kivitelezhető megoldás:

$$x_1 = 0; \quad x_2 = 0; \quad x_3 = \frac{60}{1} = 60; \quad x_4 = \frac{60}{1} = 60; \quad z = 0$$

kifejezve x_3-at a táblázat 2. sorából, x_4-et a 3. sorából.

Az optimális megoldás (helye és értéke) meghatározható pivotálás segítségével, általános kivitelezhető megoldásokkal egyre nagyobb z értékekkel, egészen addig, míg maximum z elérésre nem kerül.

A pivot egyenlet és pivot választása eltér a Gauss féle eliminációnál megismerttől, mivel x_1, x_2, x_3, x_4 nem lehetnek negatív számok.

A Simplex módszer lépései:

1. lépés, A művelet (primál oszlop kiválasztása)

   Kiválasztjuk a primál oszlopnak az első negatív értékkel bíró oszlopot az első sorból. (mi esetünkben ez a 2. lesz, -40 értékkel).

1.B Művelet: (primál sor kiválasztása)

Elosztjuk a jobb oldalakat (60 és 60) a hozzátartozó bemenetekkel a már kiválasztott oszlopból (60/2=30 és 60/5=12). Primál egyenletnek kivesszük a legkisebb kvóciensű egyenletet (mi esetünkben 5, mivel 60/5 a legkisebb).

1.C Művelet: (elimináció sorműveletekkel)

A primál alatt és fölött kinullázunk mindent.

$$T_1 = \begin{array}{c|c:ccc:c|}
\multicolumn{1}{c}{z} & \multicolumn{1}{c}{x_1} & x_2 & x_3 & \multicolumn{1}{c}{x_4} & \multicolumn{1}{c}{b} \\ \hline
1 & 0 & -72 & 0 & 8 & 480 \\ \hdashline
0 & 0 & 7{,}2 & 1 & -0{,}4 & 36 \\
0 & 5 & 2 & 0 & 1 & 60
\end{array}$$

// 1. sor + 8 * 3. sor

// 2. sor – 0,4 * 3. sor

2. Lépés

A kiszámolt általános kivitelezhető megoldás még nem optimális, mivel az 1. sorban -72 szerepel. Így végre kell hajtani az A és C műveletet még egyszer el kell végezni, primál oszlopként a -72-t kiválasztva.

2.A művelet:

Kiválasztjuk a 3. oszlopot T_1-ből primálnak.

2.B művelet:

$\frac{36}{7{,}2} = 5$ és $\frac{60}{2} = 30$ → 7,2 a primál

2.C művelet:

Sorok eliminálása:

$$T_2 = \begin{array}{c|c:ccc:c|}
\multicolumn{1}{c}{z} & \multicolumn{1}{c}{x_1} & x_2 & x_3 & \multicolumn{1}{c}{x_4} & \multicolumn{1}{c}{b} \\ \hline
1 & 0 & 0 & 10 & 4 & 840 \\ \hdashline
0 & 0 & 7{,}2 & 1 & -0{,}4 & 36 \\
0 & 5 & 0 & -\frac{1}{3{,}6} & \frac{1}{0{,}9} & 50
\end{array}$$

// 1. sor + 10* 2. sor

//3. sor – 2/7,2 * 2. sor

Vagyis x_1 és x_2 általános, x_3 és x_4 nem általános. Utóbbiakat nullázva, T_2-ből a megoldás:

$$x_1 = \frac{50}{5} = 10; \quad x_2 = \frac{36}{7{,}2} = 5; \quad x_3 = 0; \quad x_4 = 0; \quad z = 840$$

Ha minimalizálni akarunk és nem maximalizálni, akkor pozitív értéket kell venni a primálnak, nem negatívat.

Szerencsére nem mindig kell ezt a folyamatot végig csinálni, az Excel Solver-e ebben is segítségünkre van.

6.3. Példa

Feladat:

Tegyük fel, hogy fogyni szeretnénk, és optimalizálni kellene a fogyasztásunkat. Adott a következő táblázat:

	Sör	Kolbász	Pálinka	Zsemle
Kalória (cal)	380	371	260	277
Szénhidrát (g)	4,6	0	0	57,4
Fehérje (g)	0	18	0	9,2
Zsír (g)	0	32	0	0
Költség (Ft)	250	500	300	30

A célunk, hogy egy minimum költségvetésű diétát hozzunk létre legalább 1000 kalóriát fogyasztva, 20 g-nál kevesebb szénhidrátot, legalább 20 g fehérjét és legalább 8 g zsír fogyasztva!

Megoldás:

A megoldáshoz a következő lépéseket hajtjuk végre:

1. **Döntési változók megadása**

Az elején megadjuk azokat a változókat, amire végeredményben majd kíváncsiak leszünk, tehát hogy miből mennyi fogyaszthatunk. Kezdjük igazságosan mindenből 1-et!

	A	B	C	D	E
1	**Döntési változók**				
2		Sör	Kolbász	Pálinka	Zsemle
3	Fogyasztás	1	1	1	1

A végeredmény persze nem feltétlenül egész szám lesz, de kiindulásnak megteszi.

2. **Célfüggvény meghatározása**

Nincs optimalizáció célfüggvény nélkül (anélkül csak érzésre – vagy anélkül – próbálkozunk, de tudomány semmi nincs benne!), így meg kell adnunk, hogy az optimális diétának mekkora a költsége. A 7. sort csak szimplán egyenlővé tesszük a 3. sorral, míg a 8. sorba beírjuk a költségeket az eredeti táblázatból.

A 4 oszlop elemeit páronként kellene összeszorozni, amit megadhatunk akár így is:

= B7*B8 + C7*C8 +D7*D8+ E7*E8

Persze sokkal elegánsabb a következő függvényt használni:

= SZORZATÖSSZEG(B7:E7;B8:E8)

Természetesen ezt a képletet B10 cellában fogjuk használni.

5	**Célfüggvény**				
6		Sör	Kolbász	Pálinka	Zsemle
7	Fogyasztott	1	1	1	1
8	Költség	250	500	300	30
9					
10	Összesen:	1080			

Ekkor ki is jön egy tisztességes 1080 Ft-os összeg.

3. Kényszerek megadása

Ezután definiálnunk kell azt, hogy mit szeretnénk elérni a diéta során, vagyis hogy a szoftver csak olyan megoldásokat keressen, amik az általunk megadott feltételeknek eleget tesz!

12	**Kényszerek**							
13		Sör	Kolbász	Pálinka	Zsemle	Összesen		**Szükséges**
14	Kalória	380	371	260	277	1288	>=	1000
15	Szénhidrát	4,6	0	0	57,4	62	<=	20
16	Fehérje	0	18	0	9,2	27,2	>=	20
17	Zsír	0	32	0	0	32	>=	8

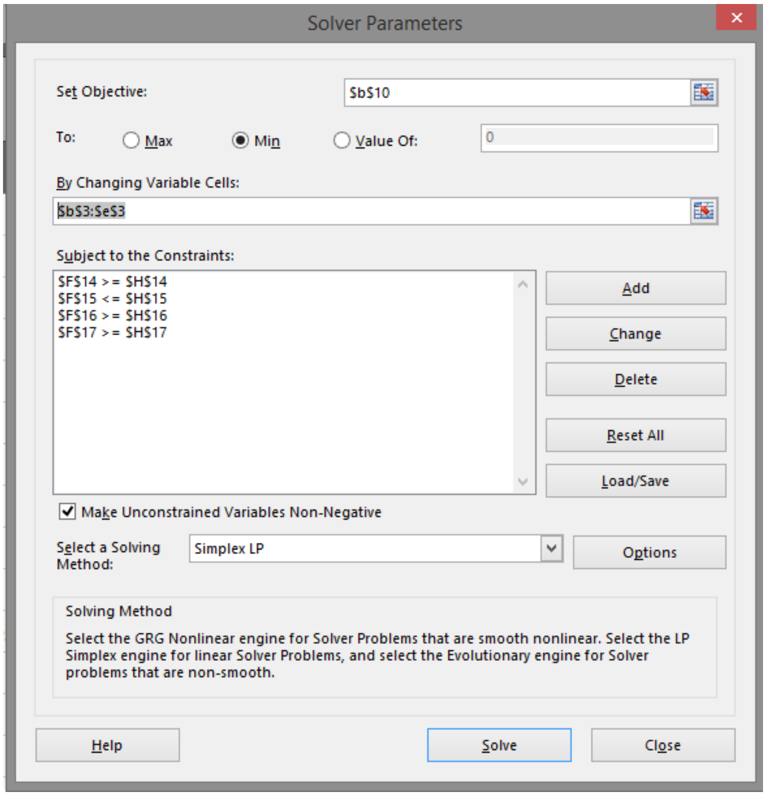
Természetesen az összesen oszlop megadása a következőképpen alakul: (F14)

=SZORZATÖSSZEG($B$7:$E$7,B14:E14)

Míg H oszlopban megadjuk a maximum (minimum) értékeket.

4. Solver beállítása

A solver ablak megnyitásával a következő dolgokat kell beállítanunk:



- Cél cella megadása: $B$10
- Cél: minimum keresés (min) – lehet maximumot, minimumot vagy éppen értéket definiálni
- Bemenő paraméterek: b3:e3 (amiket az elején megadtunk inputként)
- Kényszerek megadása: ezt a hozzáadás gombbal tudjuk megadni egyesével, ahol inputként definiáljuk F14:F17 cellákat, a logikai összefüggést, illetve célként pedig H14:H17 cellákat.
- Solver megoldó algoritmus: Simplex LP

5. Megoldás – KLIKK!

Ha mindent az előírtak szerint csináltunk, akkor a Solver jelzi, hogy talált megoldást, és ő akkor le is cserélné a meglévő számokat – azért jó fej, hogy nem csak úgy kérdés nélkül teszi meg ezt. Persze fogadjuk el, és lássuk, hogy az ideális diétánk 879 Ft-ba kerül, megihatunk 1,5 sört, és megehetünk hozzá egy negyed zsemlét. Igazi kolis diéta...

7 Referenciák

1. Faragó, I. and R. Horváth, *Numerikus Módszerek*, F. Miklós, Editor. 2011, BME TTK Matematika Intézet: Budapest.
2. Dr. Nyakóné dr. Juhász Katalin, et al., *Bevezetés az Informatikába*. 2011, Debreceni Egyetem, Informatikai Kar; Sapientia EMTE, Műszaki és Humántudományok Kar: Kempelen Farkas Hallgatói Információs Központ.
3. Esfandiari, R.S., *Numerical Methods for Engineers and Scientists Using MATLAB*. 2013: Taylor & Francis Group, LLC.

<< 10-20 soros ismertető a könyv hátsó borítójára>>

A jegyzet anyaga a BME Közlekedésmérnöki és Járműmérnöki Kar tantervéhez igazodva a Mérnöki számítások tárgy egyesített elméleti és gyakorlati segédanyaga. A jegyzet az alapvető numerikus módszerek világába ad betekintést, az elméleti hátteret Excelben illetve Excel VBA-ban készített példákon keresztül próbálja szemléltetni. A jegyzet a numerikus eljárások során felmerülő hibák, azok eredetének és végeredményére való hatásának bemutatásával indul. Ezután következnek a görbeillesztési eljárások, melyek alapvetően két fő részre bonthatók: regressziós eljárások és interpolációs módszerek. Ezután a numerikus deriválás és integrálás fejezetek következnek, végül pedig egy lineáris optimalizálási eljárás: a Simplex módszer. A jegyzetben az elméleti hátteret számos gyakorlati példa szemlélteti, melyek segítségével a tudásanyag sokkal könnyebben elsajátítható.