Írta:
MIHÁLYKÓ CSABA
VIRÁGH JÁNOS

KÖZELÍTŐ ÉS SZIMBOLIKUS SZÁMÍTÁSOK FELADATGYŰJTEMÉNY

Egyetemi tananyag

TYPOTEX

2011

LEKTORÁLTA: Dr. Molnárka Győző, Széchenyi István Egyetem Műszaki Tudományi Kar Mechatronika és Gépszerkezettan Tanszék

TÁMOGATÁS:
Készült a TÁMOP-4.1.2-08/1/A-2009-0008 számú, „Tananyagfejlesztés mérnök informatikus, programtervező informatikus és gazdaságinformatikus képzésekhez” című projekt keretében.

ISBN 978 963 279 525 6

KÉSZÜLT: a Typotex Kiadó gondozásában

FELELŐS VEZETŐ: Votisky Zsuzsa

AZ ELEKTRONIKUS KIADÁST ELŐKÉSZÍTETTE: Gerner József

KULCSSZAVAK:
numerikus matematika, feladatgyűjtemény, matematikai programcsomagok, Maple, Scilab, GSL, LAPACK, C, informatikus BSc és MSc oktatás

ÖSSZEFOGLALÁS:
A feladatgyűjtemény főleg a különböző informatika BSc és MSc szakokon tanuló hallgatók számára készült. Az első hat fejezet nagyjából a BSc, a második hat inkább az MSc szakok tematikáját dolgozza fel. Haszonnal forgathatják mindazon mérnökhallgatók is, akik a numerikus matematikához kapcsolódó tanulmányokat folytatnak.
Az 500 feladat különböző nehézségű, az egészen egyszerűtől kezdve a több ismeret összekapcsolását igénylő összetettebb problémákig. Jellegük is változatos: vannak csak számolást igénylő, begyakorlást elősegítő típusú és elmélyültebb tudást feltételező „elméletibb” feladatok is. A gyűjtemény tartalmaz olyan feladatokat is, melyek célja valamely numerikus módszer demonstrációs szintű implementálása. A mintaprogramok az első részben a Scilab 5.2 verzióját, a másodikban a Maple 13-at, illetve a (Linuxon futó) standard C nyelvet és gcc fordítóprogramot feltételeznek.

Tartalomjegyzék

Ábrák jegyzéke

Előszó

Ez a feladatgyűjtemény főleg a különböző informatika BSc és MSc szakokon tanuló hallgatók számára készült. Az első hat fejezet nagyjából a BSc, a második hat (némi átfedéssel) inkább az MSc szakok tematikáját dolgozza fel. A gyűjteményt haszonnal forgathatják mindazon mérnökhallgatók is, akik a numerikus matematikához kapcsolódó tanulmányokat folytatnak.

A feladatok különböző nehézségűek, az egészen egyszerűtől kezdve a több ismeret összekapcsolását igénylő összetettebb problémákig. Jellegük is változatos: vannak csak számolást igénylő, begyakorlást elősegítő típusú és elmélyültebb tudást feltételező elméleti(bb) feladatok is.

Terjedelmi korlátok miatt a viszonylag nagyszámú (mintegy 500) feladatnak csak egy részéhez adtunk megoldást. Reményeink szerint ezek, a rövid útmutatások, továbbá az Irodalomjegyzékben szereplő jegyzetek és szakkönyvek tanulmányozása után a többi feladat sem okoz nehézséget. A felhasznált fogalmakkal, matematikai eredményekkel, a feladatokhoz szükséges numerikus algoritmusok részleteivel kapcsolatban szintén az Irodalomjegyzék adhat eligazítást.

Amint a [2]– [6] és [15] irodalmakból is látszik, a numerikus matematika oktatásában az utóbbi években jelentős hangsúlyeltolódások történtek. A régebbi, papír+ceruza/zsebszámológép, sok elmélet, sok tétel megközelítés mellett mostanában inkább a kísérletező tanulásra buzdító, a tanulást matematikai programcsomagokkal (Matlab, Maple, Octave, Scilab, stb) megtámogató szemlélet nyert teret.

A gyűjtemény ezért is tartalmaz olyan feladatokat, melyek célja valamely numerikus módszer demonstrációs szintű implementálása. A mintaprogramok az első részben a Scilab 5.2 verzióját, a másodikban a Maple 13-at, illetve a (Linuxon futó) standard C nyelvet/gcc fordítóprogramot feltételeznek.

Ezen túl szinte minden feladat megoldásánál nagy segítség lehet, ha kézi számolás helyett valamilyen kézreálló matematikai szoftvert használ az Olvasó.

Végül szeretnénk köszönetet mondani Pozsgai Tamásnak, aki nagy segítséget nyújtott a feladatok, programok és ábrák egy részének elkészítésében, formázásában.

Szeged–Veszprém, 2010. december

Mihálykó Csaba – Virágh János

Jelölések

$\mathbb{N}, \mathbb{R}, \mathbb{C}$	a természetes, a valós, illetve a komplex számok halmaza
$\lvert a \rvert$	az a szám abszolút értéke
$\deg(p)$	a p(x) polinom fokszáma
$\mathbb{R}^n$	az n dimenziós valós vektorok halmaza
$\mathbb{R}^{n\times m}$	az $n \times m$-es valós mátrixok halmaza
$\mathbf{a}, \mathbf{b}, \ldots, \mathbf{w}$	tetszőleges vektorok
$\lVert \mathbf{a} \rVert$	az **a** vektor normája
$\mathbf{a}^{\mathrm{T}} = (a_1, a_2, \ldots, a_n)$	az **a** vektor transzponáltja, a komponenseket vessző választja el
$\langle \mathbf{a}, \mathbf{b} \rangle \in \mathbb{R}$	az **a** és **b** vektorok belső vagy skaláris szorzata
$\mathbf{a}\mathbf{b}^{\mathrm{T}} \in \mathbb{R}^{n\times n}$	az **a** és **b** vektorok külső vagy diadikus szorzata
$\mathcal{G} = \mathrm{span}(\mathbf{g}_1, \mathbf{g}_2, \ldots, \mathbf{g}_n)$	a $\mathbf{g}_1, \mathbf{g}_2, \ldots, \mathbf{g}_n$ vektorok által generált (kifeszített) altér
$\mathbf{A}, \mathbf{B}, \ldots, \mathbf{W}$	tetszőleges mátrixok
$\lVert \mathbf{A} \rVert$	az **A** mátrix normája
$\mathbf{A}^{\mathrm{T}}$	az **A** mátrix transzponáltja
$\det(\mathbf{A})$	az **A** mátrix determinánsa
$\mathbf{A}^{-1}$	az **A** mátrix inverze
$p_{\mathbf{A}}(\lambda)$	az **A** mátrix karakterisztikus polinomja
$\rho(\mathbf{A})$	az **A** mátrix spektrálsugara
$\deg(p(x))$	a $p(x)$ polinom fokszáma
$f : \mathbb{R} \mapsto \mathbb{R}$	$f(x)$ egyváltozós valós-valós függvény
$f'(x), f''(x), \ldots, f^{(k)}(x)$	az $f(x)$ függvény első, második, …, k-dik deriváltja
$f(x) \in \mathrm{C}[a, b]$	$f(x)$ $[a, b]$-n folytonos
$f(x) \in \mathrm{C}^n[a, b]$	$f(x)$ $[a, b]$-n n-szer folytonosan differenciálható függvény
$\mathbf{F} : \mathbb{R}^n \mapsto \mathbb{R}^n$	$\mathbf{F}(\mathbf{x})$ $\mathbb{R}^n$ -ből $\mathbb{R}^n$ -be leképező „vektor-vektor” függvény
$F_i : \mathbb{R}^n \mapsto \mathbb{R}$	az $\mathbf{F}(\mathbf{x})$ függvény F_i i-dik komponens függvénye

A fönti jelölésekben *halmazok* szerepelnek. Az egyszerűség kedvéért az egyes matematikai struktúrákat tartóhalmazukkal azonos módon jelöljük. Így például az $n \times n$-es valós mátrixok *gyűrűjére* szintén az $\mathbb{R}^{n\times n}$ szimbólummal hivatkozunk.

Első rész

Feladatok

1. fejezet

A numerikus analízis alapfogalmai

1.1. Egész számok ábrázolása

1.1.1. Feladat. Adja meg 8 biten direkt kód használatával az 59-et, a -59-et, a 13-at, a 42-t és a -15-öt!

1.1.2. Feladat. Adja meg 8 biten kettes komplemens kód használatával az 59-et, a -59-et, a 13-at, a 12-t és a -15-öt!

1.1.3. Feladat. Adja meg 8 biten 129-et, ha lehetséges direkt kód használatával!

1.1.4. Feladat. Legalább hány bit szükséges ahhoz, hogy 129-et ábrázolni lehessen direkt kódban? Adja meg ennyi bit segítségével a számot!

1.1.5. Feladat. Adja meg 16 biten -45629-et kettes komplemens kódot használva!

1.1.6. Feladat. Legalább hány bit szükséges ahhoz, hogy -45629-et ábrázolni lehessen kettes komplemens kódban? Adja meg ennyi bit segítségével a számot!

1.1.7. Feladat. Adja meg azokat a tízes számrendszerbeli számokat, amelynek a 8 bites alakja direkt kódban:
a) 10110011 b) 11001110 c) 01101100 d) 01001110

1.1.8. Feladat. Adja meg azokat a tízes számrendszerbeli számokat, amelynek a 8 bites alakja kettes komplemens kódban:
a) 11000011 b) 10110011 c) 01101100 d) 01000110

1.1.9. Feladat. Melyik az az x negatív egész szám, amelynek direkt kódban és kettes komplemens kódban m biten felírt alakja megegyezik!

1.1.10. Feladat. Mutassa meg, hogy ha $-2^{m-1} \leq x < 0, x \in \mathbb{Z}$, akkor ugyanazt a kettes számrendszerben felírt számot kapjuk, ha a $c = 2^m + x$ számot ábrázoljuk m biten, valamint, ha m biten ábázoljuk $|x|$-et és utána minden 1-est 0-ra, minden 0-t 1-re átváltoztatunk és végül az így kapott számhoz hozzáadunk 1-et!

1.1.11. Feladat. Mi mondható az előző feladatban megadott módon felírt két kettes számrendszerbeli szám egyenlőségéről, ha x nem a megadott tartományba esik?

1.2. Lebegőpontos számok ábrázolása

1.2.1. Feladat. Adja meg a -4.6, a 13.2, a 135.5 és a -105.25 egyszeres pontosságú lebegőpontos alakját!

1.2.2. Feladat. Adja meg a 42.8, a -8.4, a 139.25 és a -1000.5 dupla pontosságú lebegőpontos alakját!

1.2.3. Feladat. Határozza meg mivel egyenlőek tízes számrendszerben felírva az alább megadott egyszeres pontosságú lebegőpontos számok!
a) 11100000111010000000000000000000

b) 11000111010110000000000000000000

c) 01100111001101000000000000000000

1.2.4. Feladat. Adja meg, hogy mi lesz az értéke az x számnak, ha egyszeres pontossággal számolunk és az adott sorrendben végezzük el a műveleteket!

$$x = 2^{35} + 1000 - 2^{35}$$

1.2.5. Feladat. Adja meg, hogy mi lesz az értéke az x számnak, ha dupla pontossággal számolunk és az adott sorrendben végezzük el a műveleteket!

$$x = 2^{35} + 1000 - 2^{35}$$

1.2.6. Feladat. Minimálisan hány biten lehet ábrázolni a -12.25-öt, ha az ábrázolás struktúrája megegyezik az egyszeres illetve a dupla pontosságú ábrázolás struktúrájával?

1.3. Hibaanalízis

1.3.1. Feladat. Legyen $x = 5.01$, $y = -2.49$, míg közelítő értékei $\widetilde{x} = 5$, $\widetilde{y} = -2.5$. Adja meg, hogy mekkora abszolút hibát követ el, ha a pontos értékek helyett a közelítő értékeket adja össze, vonja ki, szorozza össze, illetve osztja el!

1.3.2. Feladat. Az előző feladatbeli pontos és közelítő értékekkel számolva mekkora lesz a relatív hiba az alapműveletek elvégzésekor?

1.3.3. Feladat. Adjon becslést x pontos értékére, ha tudja, hogy a közelítő értéke 5 és az ehhez tartozó abszolút hiba nem nagyobb, mint 10^{-2}!

1.3.4. Feladat. Adjon becslést x pontos értékére, ha tudja, hogy a közelítő értéke 5 és az ehhez tartozó relatív hiba nem nagyobb, mint 10^{-2}!

1.3.5. Feladat. Az előző két feladatban szereplő 10^{-2} hibakorlát helyett adható lenne-e olyan közös hibakorlát, hogy eredményül ugyanazt a becslést kapja x-re?

1.3.6. Feladat. Létezik-e olyan $\widetilde{x}$ közelítő érték, amely esetén adható közös $\epsilon > 0$ hibakorlát az abszolút és a relatív hibára az előző feladat mintájára?

1.3.7. Feladat. Létezik-e olyan $\widetilde{x}$ közelítő érték, amely esetén megadható olyan pozitív ϵ_1 illetve ϵ_2 abszolút, illetve relatív hibakorlát, hogy ugyanazt a becslést kapja a pontos x értékre?

1.3.8. Feladat. Legyen $x = 5.1$, $y = 3.05$, $\widetilde{x} = 5$, $\widetilde{y} = 3$. Adjon becslést a

$$\frac{2xy}{x-y}$$

kifejezés abszolút hibájára a számítások elvégzése nélkül. A becslés során x és y helyett használja $\widetilde{x}$-t és $\widetilde{y}$-t! Hasonlítsa össze az így kapott becslést a tényleges abszolút hibával!

1.3.9. Feladat. Az előző feladatbeli x, y, $\widetilde{x}$, $\widetilde{y}$ értékek mellett adjon becslést a $\frac{2xy}{x-y}$ kifejezés relatív hibájára a számítások elvégzése nélkül. A becslés során x és y helyett használja $\widetilde{x}$-t és $\widetilde{y}$-t!

1.3.10. Feladat. Legyenek $\widetilde{x} = 5$, $\widetilde{y} = 2$ közelítő értékek 10^{-2} abszolút hibakorláttal. Adjon becslést az $\widetilde{x} \cdot \widetilde{y} + \widetilde{x}^2$ kiszámításakor elkövetett abszolút hibára!

1.3.11. Feladat. Legyenek $\widetilde{x} = 5$, $\widetilde{y} = 2$ közelítő értékek 10^{-3} relatív hibakorláttal. Adjon becslést az $\widetilde{x} \cdot \widetilde{y} + \widetilde{x}^2$ kiszámításakor elkövetett relatív hibára!

1.3.12. Feladat. Az alábbi műveletek közül melynek lesz/lehet kisebb az abszolút hibára vonatkozó becslése? Adjon példát!

a) $\dfrac{\widetilde{x}+\widetilde{y}}{\widetilde{z}}$ b) $\dfrac{\widetilde{x}}{\widetilde{z}} + \dfrac{\widetilde{y}}{\widetilde{z}}$

1.3.13. Feladat. Vizsgálja meg az előző feladatbeli műveleteket a relatív hiba szempontjából!

1.3.14. Feladat. Adjon becslést az $(\widetilde{x}+\widetilde{y}) \cdot \widetilde{z}$ illetve az $\widetilde{x} \cdot \widetilde{z} + \widetilde{y} \cdot \widetilde{z}$ értékek számításakor fellépő abszolút hibakorlátra!

1.3.15. Feladat. Vizsgálja meg az előző feladatbeli értékeket a relatív hibakorlát szempontjából!

1.3.16. Feladat. Adja meg a pontos tizedesjegyek számát az e következő közelítése esetében: $e \approx 2.7183$!

1.3.17. Feladat. Adja meg 10 tizedesjegy pontossággal e értékét!

1.3.18. Feladat. Határozza meg, hogy legfeljebb mekkora abszolút hibát követhetünk el, ha a $\sin\left(\frac{999\pi}{1000}\right)$ értékét $\sin(\pi)$-vel helyettesítjük!

1.3.19. Feladat. Adjon becslést az előző feladatban elkövetett relatív hibára!

1.3.20. Feladat. Legfeljebb mekkora abszolút hibával rendelkező $\widetilde{x}$-mal helyettesíthetjük x-et, ha $\ln(x)$ értékét legfeljebb 10^{-3}-os abszolút hibával szeretnénk közelíteni $\ln(\widetilde{x})$-mal és x végigfut az $[1,10]$ intervallumon?

1.3.21. Feladat. Egy téglatest 3 oldalát rendre 10, 20 és 50 cm-nek mérték. Az elkövetett mérési hiba sorra legfeljebb 1, 2 és 5 mm. Mekkora lehet legfeljebb a téglatest felülete és térfogata? Adjon felső becslést a felület és a térfogat számításakor elkövetett relatív hibára!

1.3.22. Feladat. Egy mérés szerint egy test 5 másodperc alatt 10m utat tett meg. Mind az időtartam, mind a távolság mérésekor maximum 1%-os hiba történt. Mennyi volt a test átlagsebessége? Adjon az eredményre vonatkozóan abszolút és relatív hibakorlátot!

2. fejezet

A lineáris algebra numerikus módszerei

2.1. Eliminációs módszerek

2.1.1. Feladat. Oldja meg Gauss-eliminációval az alábbi egyenletrendszereket (ha lehet)!

a)
$$\begin{aligned} -2x_1-3x_2-3x_4 &= 11 \\ -3x_1-2x_2-2x_3-3x_4 &= -10 \\ 3x_1-3x_2+7x_3+2x_4 &= 66 \\ -x_1+x_2-2x_3+x_4 &= -22 \end{aligned}$$

b)
$$\begin{aligned} -3x_1-2x_2+x_3 &= 8 \\ x_1-2x_2-x_3 &= -24 \\ 12x_1-2x_2-x_3 &= -86 \end{aligned}$$

c)
$$\begin{aligned} 3x_2+x_3 &= 1 \\ -2x_1+10x_2+2x_3 &= 2 \\ -3x_1-x_2+2x_3 &= -24 \end{aligned}$$

d)
$$\begin{aligned} x_1+2x_2+3x_3 &= 12 \\ 4x_1+5x_2+6x_3 &= 30 \\ 7x_1+8x_2+9x_3 &= 48 \end{aligned}$$

e)
$$\begin{aligned} 12x_1-2x_2-x_3 &= 13 \\ -x_1+2x_2+3x_3 &= 4 \\ 6x_1+4x_2-2x_3 &= -8 \end{aligned}$$

f)
$$\begin{aligned} x_1+8x_2+3x_3 &= 64 \\ 11x_1+4x_2+10x_3 &= 32 \\ 12x_1+11x_2+8x_3 &= 88 \end{aligned}$$

g)
$$\begin{aligned} -2x_1+4x_2+6x_3 &= 8 \\ 12x_1-2x_2-x_3 &= 13 \\ -3x_1-2x_2+x_3 &= 4 \end{aligned}$$

h)
$$\begin{aligned} 4x_1+2x_2+x_3 &= 2 \\ 2x_1+2x_2+x_3 &= 0 \\ x_1+x_2+\frac{5}{2}x_3 &= 0 \end{aligned}$$

2.1.2. Feladat. Oldja meg a 2.1.1. feladat egyenletrendszereit Gauss-Jordan módszerrel (ha lehet)!

2.1.3. Feladat. Oldja meg a 2.1.1. feladat egyenletrendszereit oszlop szerinti részleges főelem-kiválasztásos Gauss-eliminációval!

2.1.4. Feladat. Oldja meg a 2.1.1. feladat egyenletrendszereit teljes főelemkiválasztásos Gauss-eliminációval!

2.1.5. Feladat. Határozza meg a 2.1.1. feladat egyenletrendszereiben szereplő együttható mátrixok determinánsát a Gauss-eliminációt felhasználva! Ahol szükséges, alkalmazzon sorcserét!

2.1.6. Feladat. Adja meg a 2.1.1.a. és a 2.1.1.b. feladatbeli egyenletrendszerekben szereplő együttható mátrixok LU felbontását! Ezek után oldja meg az egyenletrendszereket az L és az U mátrixok segítségével!

2.1.7. Feladat. Mutassa meg, hogy az alábbi mátrixnak nem létezik LU felbontása!

$$\mathbf{A} = \begin{pmatrix} 4 & 4 & 2 \\ 1 & 1 & 1 \\ 8 & 1 & 0 \end{pmatrix}$$

2.1.8. Feladat. Mutassa meg, hogy az alábbi mátrixnak végtelen sok LU felbontása van!

$$\mathbf{A} = \begin{pmatrix} 1 & 1 & 1 \\ 3 & 3 & 1 \\ 82 & 2 & 1 \end{pmatrix}$$

2.1.9. Feladat. Oldja meg a 2.1.1.h. feladatbeli egyenletrendszert Cholesky felbontás segítségével!

2.1.10. Feladat. Mutassa meg, hogy ha az **A** mátrixnak létezik Cholesky felbontása, akkor

$$a_{ii} \geq 0 \qquad \forall i\text{-re.}$$

2.1.11. Feladat. Mutassa meg, hogy az alábbi mátrixnak nem létezik Cholesky felbontása!

$$\mathbf{A} = \begin{pmatrix} 0 & 2 \\ 2 & 0 \end{pmatrix}$$

2.1.12. Feladat. Invertálja a következő mátrixokat!

a) $$\mathbf{A} = \begin{pmatrix} 1 & -2 & 3 & 1 \\ -2 & 1 & -2 & -1 \\ 3 & -2 & 1 & 5 \\ 1 & -1 & 5 & 3 \end{pmatrix}$$ b) $$\mathbf{A} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 3 \\ 1 & 3 & 6 \end{pmatrix}$$

2.1.13. Feladat. Igaz-e, hogy trianguláris, tridiagonális illetve szalagmátrix inverze is trianguláris, tridiagonális illetve szalagmátrix?

2.1.14. Feladat. Határozza meg az alábbi **A** mátrixban α és β értékét úgy, hogy a mátrix invertálható legyen és adja meg $\mathbf{A}^{-1}$-t!

$$\mathbf{A} = \begin{pmatrix} \alpha & 1 & 0 \\ \beta & 2 & 1 \\ 0 & 1 & 2 \end{pmatrix}$$

2.1.15. Feladat. A 2.1.14. feladatbeli **A** mátrixban határozza meg α és β értékét úgy, hogy a mátrix legyen

a) soronként domináns diagonálisú, b) oszloponként domináns diagonálisú.

Mutassa meg, hogy mindkét esetben invertálható az **A** mátrix!

2.1.16. Feladat. A 2.1.14. feladatbeli **A** mátrixban határozza meg α és β értékét úgy, hogy a mátrix pozitív definit legyen!

2.1.17. Feladat. Bizonyítsa be, hogy tetszőleges, akár soronként, akár oszloponként domináns diagonálisú mátrix invertálható!

2.1.18. Feladat. Határozza meg b értékét úgy, hogy legyen megoldása az alábbi lineáris egyenletrendszernek! Adja meg a megoldást is!

$$\begin{aligned} x_1+2x_2+x_3 &= 1 \\ 2x_1+4x_2+8x_3 &= b \\ 6x_1+12x_2+16x_3 &= -4 \end{aligned}$$

2.1.19. Feladat. Határozza meg c értékét az alábbi egyenletrendszerben úgy, hogy 0, 1 illetve végtelen sok megoldás legyen! Adja meg a megoldásokat!

$$\begin{aligned} x_1+cx_3 &= 5 \\ x_1+x_2+x_3 &= 4 \\ cx_1+x_3 &= c+4 \end{aligned}$$

2.1.20. Feladat. Készítsen olyan (Scilab) eljárást, amely adott lineáris egyenletrendszert old meg Gauss eliminációval.
Az eljárás paraméterezése `Gauss(A,b)` alakú legyen, ahol

`Gauss`	az eljárás neve,
`A`	az egyenletrendszer bal oldalának együtthatóiból képzett mátrix,
`b`	az egyenletrendszer jobb oldalából képzett oszlopvektor.

2.1.21. Feladat. Készítsen olyan (Scilab) eljárást, amely adott lineáris egyenletrendszert old meg Gauss-Jordan eliminációval.
Az eljárás paraméterezése `GaussJordan(A,b)` alakú legyen, ahol

`GaussJordan`	az eljárás neve,
`A`	az egyenletrendszer bal oldalának együtthatóiból képzett mátrix,
`b`	az egyenletrendszer jobb oldalából képzett oszlopvektor.

2.1.22. Feladat. Készítsen olyan (Scilab) eljárást, amely adott lineáris egyenletrendszert old meg oszlop szerinti részleges főelemkiválasztásos Gauss eliminációval.
Az eljárás paraméterezése `rGauss(A,b)` alakú legyen, ahol

`rGauss`	az eljárás neve,
`A`	az egyenletrendszer bal oldalának együtthatóiból képzett mátrix,
`b`	az egyenletrendszer jobb oldalából képzett oszlopvektor.

2.1.23. Feladat. A 2.1.20. feladatban elkészített programmal oldja meg a 2.1.1. feladat egyenletrendszereit!

2.1.24. Feladat. A 2.1.21. feladatban elkészített programmal oldja meg a 2.1.1. feladat egyenletrendszereit!

2.1.25. Feladat. A 2.1.22. feladatban elkészített programmal oldja meg a 2.1.1. feladat egyenletrendszereit!

2.2. Vektor- és mátrixnormák, Jacobi és Gauss-Seidel iteráció

2.2.1. Feladat. Határozza meg az alábbi vektorok 1-es, 2-es és végtelen normáját!

a) $\mathbf{v} = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$ b) $\mathbf{v} = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$ c) $\mathbf{v} = (1, 2, 3, 4)^T$

2.2.2. Feladat. Bizonyítsa be, hogy az alábbiak vektornormák!

a) $||\mathbf{x}|| = max_{1\le j\le n}\left|\sum_{i=1}^{j} x_i\right|$ b) $||\mathbf{x}||_{\mathbf{A}} = ||\mathbf{A}\cdot\mathbf{x}||$,

ahol $||.||$ tetszőleges, adott vektornorma, $\mathbf{A}$ pedig tetszőleges, rögzített, invertálható mátrix.

2.2.3. Feladat. Bizonyítsa be, hogy tetszőleges $\mathbf{x}\in\mathbb{R}^n$ esetén igaz:
a) $||\mathbf{x}||_\infty \le ||\mathbf{x}||_1 \le n||\mathbf{x}||_\infty$
b) $||\mathbf{x}||_\infty \le ||\mathbf{x}||_2 \le \sqrt{n}||\mathbf{x}||_\infty$
c) $||\mathbf{x}||_2 \le ||\mathbf{x}||_1 \le \sqrt{n}||\mathbf{x}||_2$

2.2.4. Feladat. Bizonyítsa be, hogy tetszőleges rögzített $\mathbf{A}$ pozitív definit mátrix esetén az $||\mathbf{x}|| = \sqrt{\mathbf{x}^T A\mathbf{x}}$ vektornorma! Igaz-e ez, ha $\mathbf{A}$ pozitív szemidefinit mátrix?

2.2.5. Feladat. Bizonyítsa be, hogy tetszőleges $||.||$ vektornormára igaz, hogy $||.||$ folytonos függvény $\mathbb{R}^n$-en!

2.2.6. Feladat. Határozza meg az alábbi mátrixok 1-es, 2-es és végtelen normáját!

a) $\mathbf{A} = \begin{pmatrix} 2 & 1 \\ -1 & 3 \end{pmatrix}$ b) $\mathbf{A} = \begin{pmatrix} 3 & 0 & 0 \\ 0 & 5 & 5 \\ 0 & 5 & 5 \end{pmatrix}$

2.2.7. Feladat. Bizonyítsa be, hogy tetszőleges$||.||$ mátrixnorma esetén igaz az alábbi egyenlőtlenség!

$$|||A|| - ||B||| \le ||A - B||$$

2.2.8. Feladat. Bizonyítsa be, hogy tetszőleges $\mathbf{A}\in\mathbb{R}^{n\times n}$ esetén igaz:
a) $||A||_\infty \le n||A||_1$ b) $||A||_1 \le n||A||_\infty$

2.2.9. Feladat. Határozza meg, hogy a $\mathbf{v}_k = \left(2, \frac{1}{k}, \left(\frac{2}{3}\right)^k\right)^T$, $k = 1, 2, \ldots$ vektorsorozat hányadik eleme fog először 10^{-1}-nél kevesebbel eltérni a limeszvektortól, ha

a) 1-es b) 2-es c) végtelen

normában számolunk!

2.2.10. Feladat. Legyen $\mathbf{A}_k = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} \\ 0 & \frac{1}{2} \end{pmatrix}^k$, $k = 1, 2, 3, \ldots$. Határozza meg az $\mathbf{A}_\infty = \lim_{k\to\infty} A_k$ mátrixot! Milyen k esetén lesznek először igazak az alábbi egyenlőtlenségek?

a)
$$||A_\infty - A_k||_1 \leq 10^{-2}$$

b)
$$||A_\infty - A_k||_\infty \leq 10^{-2}$$

2.2.11. Feladat. Mutassa meg, hogy ha A oszloponként domináns diagonálisú, akkor a Jacobi-iteráció és a Gauss-Seidel iteráció is konvergens!

2.2.12. Feladat. Mutassa meg, hogy a fenti állítás akkor is igaz, ha A soronként domináns diagonálisú!

2.2.13. Feladat. Bizonyítsa be, hogy ha az A együttható mátrix invertálható és felső háromszög mátrix, akkor a Jacobi-iterációval véges sok lépésben megkapjuk az $\mathbf{Ax} = \mathbf{b}$ egyenletrendszer pontos megoldását!

2.2.14. Feladat. Bizonyítsa be, hogy a fenti állítás igaz Gauss-Seidel iteráció esetén is!

2.2.15. Feladat. Oldja meg az alábbi egyenletrendszereket Jacobi-iterációval közelítőleg úgy, hogy megálláskor az eltérés az előző vektortól végtelen normában kisebb legyen, mint $\epsilon = 5 \cdot 10^{-1}$! Az $\mathbf{x}_1$ kezdővektor legyen a nullvektor!

a)
$$\begin{aligned} 2x_1 + x_2 &= 5 \\ x_1 + 4x_2 &= 6 \end{aligned}$$

b)
$$\begin{aligned} 5x_1 + x_2 &= 5 \\ x_1 + 5x_2 &= 1 \end{aligned}$$

c)
$$\begin{aligned} 10x_1 + 2x_2 + x_3 &= 1 \\ 5x_2 + x_3 &= 1 \\ x_1 + 2x_2 + 10x_3 &= 5 \end{aligned}$$

d)
$$\begin{aligned} x_1 + 5x_2 + x_3 &= 1 \\ 5x_1 + x_3 &= 0 \\ x_1 + x_2 + 10x_3 &= 0 \end{aligned}$$

e)
$$\begin{aligned} 7x_1 - 2x_2 + 3x_3 &= 12 \\ -2x_1 - 5x_2 - x_3 + x_4 &= -10 \\ 4x_1 - 2x_2 + 10x_3 + 3x_4 &= 3 \\ -x_2 + 4x_3 - 7x_4 &= 6 \end{aligned}$$

f)
$$\begin{aligned} -4x_1 - 2x_2 + x_3 &= -9 \\ 5x_1 - 10x_2 + 2x_3 &= 8 \\ 2x_1 - 3x_2 + 7x_3 &= -3 \end{aligned}$$

2.2.16. Feladat. Oldja meg a fenti egyenletrendszereket Gauss-Seidel iterációval közelítőleg úgy, hogy megálláskor az eltérés az előző vektortól végtelen normában kisebb legyen, mint $\epsilon = 5 \cdot 10^{-1}$! Az $\mathbf{x}_1$ kezdővektor legyen a nullvektor.

2.2.17. Feladat. Adjon felső becslést a 100. közelítő vektor és a megoldásvektor eltérésére egyes normában a 2.2.15.a. és 2.2.15.b. feladatbeli egyenletrendszerek Jacobi- illetve Gauss-Seidel iterációval történő megoldásakor nullvektorból való indulással!

2.2.18. Feladat. Készítsen olyan (Scilab) eljárást, amely adott lineáris egyenletrendszer közelítő megoldását adja meg Jacobi-iterációval.
Az eljárás paraméterezése `Jacobi(A,b,x,t)` alakú legyen, ahol

`Jacobi`	az eljárás neve,
`A`	az egyenletrendszer bal oldalának együtthatóiból képzett mátrix,
`b`	az egyenletrendszer jobb oldalából képzett oszlopvektor,
`x`	kezdeti vektor,
`t`	a megoldás pontossága.

2.2.19. Feladat. Készítsen olyan (Scilab) eljárást, amely adott lineáris egyenletrendszer közelítő megoldását adja meg Gauss-Seidel iterációval.
Az eljárás paraméterezése `GaussSeidel(A,b,x,t)` alakú legyen, ahol

`GaussSeidel`	az eljárás neve,
`A`	az egyenletrendszer bal oldalának együtthatóiból képzett mátrix,
`b`	az egyenletrendszer jobb oldalából képzett oszlopvektor,
`x`	kezdeti vektor,
`t`	a megoldás pontossága.

2.2.20. Feladat. A 2.2.18. feladatban elkészített programmal oldja meg a 2.2.15.a. feladatot $\epsilon = 10^{-3}$ esetén!

2.2.21. Feladat. A 2.2.19. feladatban elkészített programmal oldja meg a 2.2.15.a. feladatot $\epsilon = 10^{-3}$ esetén!

2.3. Kondíciószám, perturbáció

2.3.1. Feladat. Számítsa ki az alábbi mátrixok $cond_1$ és $cond_\infty$ kondíciószámát!

a) $\mathbf{A} = \begin{pmatrix} 2 & 3 \\ 1 & 4 \end{pmatrix}$ b) $\mathbf{A} = \begin{pmatrix} 5 & 1 \\ 1 & 5 \end{pmatrix}$

c) $\mathbf{A} = \begin{pmatrix} 40 & 1 \\ 241 & 6 \end{pmatrix}$ d) $\mathbf{A} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 8 \\ 1 & 3 & 6 \end{pmatrix}$

2.3.2. Feladat. Tekintsük a

$$\begin{aligned} 40x_1 + x_2 &= 41 \\ 241x_1 + 6x_2 &= 247 \end{aligned}$$

egyenletrendszer következő perturbációját:

$$\begin{aligned} 40.01x_1 + x_2 &= 41 \\ 241x_1 + 6x_2 &= 247 \end{aligned}$$

Magyarázza meg a pontos megoldások nagymértékű eltérését!

2.3.3. Feladat. Vizsgálja meg, mi a helyzet, ha az előző feladatbeli első egyenletrendszer mátrixa változatlan marad, de a jobb oldal a $\widetilde{b}=(41,\, 247.01)^T$ vektorra változik! Adjon magyarázatot a megoldások nagymértékű eltérésére!

2.3.4. Feladat. Bizonyítsa be, hogy ha a jobb oldal relatív hibája 10^{-t} és az együtthatómátrix kondíciószáma 10^s, akkor s adja azon decimális számjegyek számát, amelyeket elveszítünk, ha a hibás jobb oldalhoz adjuk meg a megoldást!

2.3.5. Feladat. Mutassa meg, hogy $\mathbf{A}$ invertálható mátrix esetén a mátrix kondíciószámának értéke független a mátrix determinánsának értékétől!

3. fejezet

Függvényközelítések

3.1. Lagrange interpoláció

3.1.1. Feladat. Határozza meg a másodfokú Lagrange interpolációhoz tartozó alábbi Vandermonde mátrix determinánsát!

$$V = \begin{pmatrix} x_0^2 & x_0 & 1 \\ x_1^2 & x_1 & 1 \\ x_2^2 & x_2 & 1 \end{pmatrix}$$

3.1.2. Feladat. Határozza meg a Vandermonde együttható-mátrixú egyenletrendszer megoldásával azt a harmadfokú polinomot, amely az $x_i = i$, $i = 0,1,2,3$ pontokban felveszi az $f(i) = \sum_{j=0}^{i} j(j+1)$ értékeket! Bizonyítsa be, hogy a kapott $L(x)$ polinom minden $i \in \mathbb{N}$ esetén megegyezik $f(i)$-vel!

3.1.3. Feladat. Bizonyítsa be, hogy tetszőleges $n \in \mathbb{N}$ esetén a

$$V = \begin{pmatrix} x_0^n & x_0^{n-1} & \cdots & x_0 & 1 \\ x_1^n & x_1^{n-1} & \cdots & x_1 & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ x_n^n & x_n^{n-1} & \cdots & x_n & 1 \end{pmatrix}$$

Vandermonde mátrix pontosan akkor invertálható, ha az $x_i \in \mathbb{R}$, $i = 0,1,\ldots,n$ számok páronként különbözőek! Adja meg V determinánsát!

3.1.4. Feladat. Számítsa ki az alábbi adatokhoz tartozó Lagrange-féle interpolációs polinomokat!

a) (0,0), (1,1), (2,4) b) (0,0), (1,1), (2,8), (3,27)

3.1.5. Feladat. Adjon módszert a Lagrange interpoláció segítségével a $p_{k+1}(n) = \sum_{j=0}^{n} j^k$ polinomok meghatározására!

3.1.6. Feladat. Határozza meg a 3.1.5. feladatbeli $p_{k+1}(n)$ polinomokat $k = 0,1,2,3$ esetén!

3.1.7. Feladat. Határozza meg az $f(x)=e^x$ függvényhez tartozó n-ed fokú Lagrange interpolációs polinomot $n = 1,2,3$ esetén, ahol $x_i = \dfrac{i}{n}$, $i = 0,1,\ldots,n$.

3.1.8. Feladat. Adjon felső becslést az előző feladatban meghatározott $L_n(x)$ Lagrange interpolációs polinom és az e^x függvény eltérésére a $[0,1]$ intervallumon!

3.1.9. Feladat. Hányadfokú Lagrange interpolációs polinom szükséges ahhoz, hogy a $[0,1]$ intervallumon ekvidisztáns pontokon 10^{-3}-os pontossággal interpoláljuk az e^x függvényt?

3.1.10. Feladat. Lagrange interpolációs polinom felhasználásával közelítse a $\sin(x)$ függvényt $\mathbb{R}$-n 10^{-3} pontossággal! Hányadfokú polinom szükséges ehhez?

3.1.11. Feladat. Oldja meg az előző feladatot $\cos(x)$ esetében is!

3.1.12. Feladat. Adja meg a 3.1.10. feladatbeli Lagrange interpolációs polinomot!

3.1.13. Feladat. Mutassa meg, hogy $\forall x \in \mathbb{R}$, $n \in \mathbb{N}$ esetén a Lagrange-féle interpolációs polinom segédpolinomjainak összege 1, azaz

$$\sum_{i=0}^{n} l_i(x) = \sum_{i=0}^{n} \prod_{j=0,\, j\neq i}^{n} \frac{x - x_j}{x_i - x_j} = 1!$$

3.1.14. Feladat. Készítsen olyan (Scilab) eljárást, amely Vandermonde mátrixszal történő interpolációt valósít meg!
Az eljárás paraméterezése `LagrIntV(x,y)` alakú legyen, ahol

`LagrIntV`	az eljárás neve,
`x`	vektor az adott pontok x koordinátáiból áll,
`y`	vektor az adott pontok y koordinátáiból áll.

3.1.15. Feladat. Készítsen olyan (Scilab) eljárást, amely a Lagrange alakú polinom előállítását valósítja meg!
Az eljárás paraméterezése `LagrIntp(x,y)` alakú legyen, ahol

`LagrIntp`	az eljárás neve,
`x`	vektor az adott pontok x koordinátáiból áll,
`y`	vektor az adott pontok y koordinátáiból áll.

3.1.16. Feladat. A 3.1.14. feladatban szereplő program segítségével adja meg a 3.1.2., a 3.1.4., a 3.1.6., a 3.1.7. és a 3.1.10. feladatok polinomjait!

3.1.17. Feladat. A 3.1.15. feladatban szereplő program segítségével adja meg a 3.1.2., a 3.1.4., a 3.1.6., a 3.1.7. és a 3.1.10. feladatok polinomjait!

3.1.18. Feladat. Számítsa ki a következő osztott differenciákat!

$$f\,[x_0, x_1, \ldots, x_k]\,,\ x_k = k,\ k = 0, 1 \ldots \text{ és } f(x) = x^4.$$

Mutassa meg, hogy $x^4 = \sum_{k=0}^{4} f\,[x_0, x_1, \ldots, x_k] \prod_{j=0}^{k-1} (x - x_j)$!

3.1.19. Feladat. Vizsgálja meg, hogy mi mondható, ha az előző feladatban $f(x) = x^n$?

3.1.20. Feladat. Legyenek $x_0, x_1, \ldots, x_m$ páronként különböző valós számok. Mutassa meg, hogy ha P egy n-ed fokú polinom, akkor $P\,[x_0, x_1, \ldots, x_m] = 0$ minden $n < m$ esetén!

3.1.21. Feladat. Számítsa ki az $f[x_0, x_1, \ldots, x_k]$ osztott differenciákat, ahol $x_k = \frac{\pi}{6}k$, $k = 0, 1, \ldots, 3$ és $f(x) = \sin(x)$! Ábrázolja valamilyen ismert matematikai program segítségével az $N_n(x) = \sum_{k=0}^{n} f[x_0, x_1, \ldots, x_k] \prod_{j=0}^{k-1}(x - x_j)$, $n = 1, 2, 3$ polinomokat és a $\sin(x)$ függvényt egy ábrán a $\left[0, \frac{\pi}{2}\right]$ intervallumon!

3.1.22. Feladat. Határozza meg osztott differenciák segítségével a Newton alakját az interpolációs polinomnak a 3.1.4. feladatbeli adatok alapján!

3.1.23. Feladat. Készítsen olyan (Scilab) eljárást, amely a Newton alakú polinom előállítását valósítja meg!
Az eljárás paraméterezése `NewtIntp(x,y)` alakú legyen, ahol

`NewtIntp`	az eljárás neve,
`x`	vektor az adott pontok x koordinátáiból áll,
`y`	vektor az adott pontok y koordinátáiból áll.

3.1.24. Feladat. A 3.1.23. feladatban szereplő program segítségével adja meg a 3.1.2., a 3.1.4., a 3.1.6., a 3.1.7., a 3.1.9. és a 3.1.10. feladatok polinomjait!

3.1.25. Feladat. Vizsgálja meg az elkészített programmal az $f(x) = \frac{1}{1+25x^2}$ függvény $[-1, 1]$ intervallumon történő $N_n(x)$ Newton alakú interpolációját ekvidisztáns osztópontokon különböző $10 \leq n \leq 40$ esetén!

3.2. Hermite interpoláció

3.2.1. Feladat. Legyen f folytonosan differenciálható függvény. Mutassa meg, hogy az alábbi határértékek léteznek!
a) $f[x_0, x_0] = \lim_{x \to x_0} f[x_0, x]$ b) $f[x_0, x_0, x_1] = \lim_{x \to x_0} f[x_0, x, x_1]$
c) $f[x_0, x_1, x_1] = \lim_{x \to x_1} f[x_0, x, x_1]$
Igazolja, hogy $f[x_0, x_0] = f'(x_0)$!

3.2.2. Feladat. Adja meg az alábbi pontokon interpoláló Hermite interpolációs polinomokat!

a)

x_i	-1	1	2
y_i	1	-1	-29
y_i'	5	-7	-61

b)

x_i	0	1
y_i	0	1
y_i'	0	3

3.2.3. Feladat. Tekintse az $f(x) = \log_3(x)$ függvényt és az $x_i = 1, 3, 9, 27$ alappontokat. Határozza meg az ehhez tartozó $H_7(x)$ Hermite interpolációs polinomot! Számoljon először 4, majd 8 tizedesjegyre való kerekítéssel. Mit tapasztal?

3.2.4. Feladat. Adjon az előző feladatbeli $H_7(x)$ polinom és a $\log_3(x)$ függvény eltérésére felső becslést a 2, a 4 illetve a 20 pontokban! Ábrázolja a két függvény különbségét egy ismert matematikai szoftver segítségével! Mit mondhat a tényleges eltérésre ezekben a pontokban?

3.2.5. Feladat. Készítsen olyan (Scilab) eljárást, amely a Hermite polinom előállítását valósítja meg!
Az eljárás paraméterezése `Hermint(x,y,yd)` alakú legyen, ahol

`Hermint`	az eljárás neve,
`x`	vektor az adott pontok x koordinátáiból áll,
`y`	vektor az adott pontok y koordinátáiból áll,
`xd`	vektor az adott pontok y derivált koordinátáiból áll.

3.2.6. Feladat. A 3.2.5. feladatban szereplő program segítségével adja meg a 3.2.2. és a 3.2.3. feladatbeli polinomokat!

3.2.7. Feladat. Vizsgálja meg a 3.2.5. feladatban elkészített programmal az $f(x) = \dfrac{1}{1+25x^2}$ függvény $[-1,1]$ intervallumon történő $H_{2n+1}(x)$ Hermite interpolációját ekvidisztáns osztópontokon különböző $5 \leq n \leq 20$ esetén!

3.2.8. Feladat. Bizonyítsa be, hogy ha P egy legfeljebb $(2n+1)$-edfokú polinom, x_i $(i=0,1,\ldots,n)$ páronként különböző alappontok, akkor az alappontokhoz és a $P(x_i)$, $P'(x_i)$ értékekhez tartozó Hermite polinom azonos P-vel!

3.3. Spline interpoláció

3.3.1. Feladat. Adja meg a $\left(\dfrac{\pi}{6}i, \sin\left(\dfrac{\pi}{6}i\right)\right)$, $i=0,1,2,3$ adatokat interpoláló lineáris spline függvényt a $\left[0, \dfrac{\pi}{2}\right]$ intervallumon!

3.3.2. Feladat. Adjon felső becslést a 3.3.1. feladatbeli lineáris spline és a $\sin(x)$ függvény legnagyobb eltérésére a $\left[0, \dfrac{\pi}{2}\right]$ intervallumon!

3.3.3. Feladat. Mutassa meg, hogy egy adott $f : [a,b] \mapsto \mathbb{R}$ folytonosan differenciálható függvényt az $[a,b]$ intervallumon ekvidisztáns, h lépésközű osztópontjaiban interpoláló S_h lineáris spline függvényre igaz a következő egyenlőtlenség:

$$|f(x) - S_h(x)| \leq M_1 h, \;\; x \in [a,b]\,, \;\text{ahol}\; M_1 = \max_{x\in[a,b]} |f'(x)|$$

3.3.4. Feladat. Adja meg a $(0,0)$, $(1,1)$ és $(2,8)$ pontokat interpoláló természetes köbös spline-t!

3.3.5. Feladat. Adja meg a $(0,0)$, $(1,1)$, $(2,8)$ és $(3,27)$ pontokat interpoláló köbös spline-t!

3.3.6. Feladat. Adja meg a $\left(\dfrac{\pi}{3}i, \sin(\dfrac{\pi}{3}i)\right)$, $i=0,1,2,3$ adatokat interpoláló természetes köbös spline-t a $[0,\pi]$ intervallumon!

3.3.7. Feladat. Adjon felső becslést a 3.3.6. feladatbeli természetes köbös spline és a $\sin(x)$ függvény legnagyobb eltérésére a $[0,\pi]$ intervallumon!

3.3.8. Feladat. Készítsen olyan (Scilab) eljárást, amely a lineáris spline előállítását valósítja meg!
Az eljárás paraméterezése `LinSpl(x,y)` alakú legyen, ahol

`LinSpl`	az eljárás neve,
`x`	vektor az adott pontok x koordinátáiból áll,
`y`	vektor az adott pontok y koordinátáiból áll.

3.3.9. Feladat. A 3.3.8. feladatban elkészített program segítségével adja meg a 3.3.1. feladatbeli lineáris spline függvényt!

3.3.10. Feladat. Készítsen olyan (Scilab) eljárást, amely a természetes köbös spline előállítását valósítja meg!
Az eljárás paraméterezése `QSpl(x,y)` alakú legyen, ahol

`QSpl`	az eljárás neve,
`x`	vektor az adott pontok x koordinátáiból áll,
`y`	vektor az adott pontok y koordinátáiból áll.

3.3.11. Feladat. A 3.3.10. feladatban elkészített program segítségével adja meg a 3.3.4., a 3.3.5. és a 3.3.6. feladatokban szereplő természetes köbös spline-okat!

3.3.12. Feladat. Vizsgálja meg a 3.3.10. feladatban elkészített programmal az $f(x)=\frac{1}{1+25x^2}$ függvény $[-1,1]$ intervallumon történő természes köbös spline intepolációját ekvidisztáns osztópontokon különböző $5 \le n \le 40$ esetén!

4. fejezet

Nemlineáris egyenletek megoldása

4.1. Intervallumfelezés, húrmódszer

4.1.1. Feladat. Adja meg a következő egyenletek gyökeit az intervallumfelezéses módszer segítségével $\epsilon = 10^{-2}$-os abszolút hibakorláttal! Ahol több gyök van, ott mindegyiket határozza meg!

a)	$e^x - 5x = 0$	b)	$3x^3 - x^2 + 2x - 2 = 0$
c)	$x^3 - 3x - 2 = 0$	d)	$x + 0.5\cos(0.25x) = 0$
e)	$\ln(x) + 2 - x = 0$	f)	$e^x + x = 0$

4.1.2. Feladat. Oldja meg az előző feladat egyenleteit a húrmódszer segítségével úgy, hogy a behelyettesítési érték abszolút értéke legyen kisebb, mint 10^{-2}!

4.1.3. Feladat. Bizonyítsa be, hogy a húrmódszer konvergens a 4.1.1.a. és a 4.1.1.c. feladatok esetében a $[0,2.5]$ kezdőintervallum, illetve a 4.1.1.d. és 4.1.1.f. feladatok esetében a $[-1,0]$ kezdőintervallum választással!

4.1.4. Feladat. Az előző feladatban megadott kezdőintervallumból kiindulva határozza meg a 4.1.1.a., a 4.1.1.c., a 4.1.1.d. és 4.1.1.f. feladatbeli egyenletek esetében az adott kezdőintervallumbeli gyököt 10^{-3} abszolút hibakorláttal a húrmódszer segítségével!

4.1.5. Feladat. Adjon meg a 4.1.3. feladathoz hasonlóan olyan kezdőintervallumot, hogy a húrmódszer konvergens legyen a 4.1.1.b. és a 4.1.1.e. feladatok esetében! Igazolja a választás helyességét!

4.1.6. Feladat. Adjon becslést arra, hogy legalább hány intervallumfelezés szükséges ahhoz, hogy 10^{-3}-os abszolút hibakorláttal határozzuk meg a gyököt a 4.1.1.a. és a 4.1.1.b. feladatok esetében, ha $[0,2]$ a kezdőintervallum!

4.1.7. Feladat. Készítsen (Scilab) eljárást az intervallumfelezéses módszer előállítására! Az eljárás paraméterezése `Intfel(f,a,b,t)` alakú legyen, ahol

`Intfel`	az eljárás neve,
`f`	a függvény, aminek a gyökét keressük,
`a`	a kezdeti intervallum bal oldali végpontja,
`b`	a kezdeti intervallum jobb oldali végpontja,
`t`	a megoldás pontossága.

4.1.8. Feladat. Készítsen (Scilab) eljárást a húrmódszer előállítására!
Az eljárás paraméterezése `Hur(f,a,b,t)` alakú legyen, ahol

`Hur`	az eljárás neve,
`f`	a függvény, aminek a gyökét keressük,
`a`	a kezdeti intervallum bal oldali végpontja,
`b`	a kezdeti intervallum jobb oldali végpontja,
`t`	a megoldás pontossága.

4.1.9. Feladat. Oldja meg a 4.1.1. feladat egyenleteit a 4.1.7. feladatban elkészített programmal!

4.1.10. Feladat. Oldja meg a 4.1.1. feladat egyenleteit a 4.1.8. feladatban elkészített programmal!

4.1.11. Feladat. Vizsgálja meg az elkészített programok segítségével, hogy a 4.1.1. feladat egyenletei esetében melyik módszer bizonyul hatékonyabbnak a szükséges időigény szempontjából!

4.1.12. Feladat. Adjon meg olyan kezdőintervallumot, amelyről indítva a húrmódszer konvergál a $\sin(x)-\ln(x)=0$ egyenlet gyökéhez! Határozza is meg ezt a gyököt 10^{-3}-os abszolút hibakorláttal!

4.1.13. Feladat. Határozza meg közelítőleg az $f(x)=0$ egyenletek x^* gyökét intervallumfelezéssel oly módon, hogy a közelítő megoldásként elfogadott x_n-re teljesüljenek az alábbiak:

$$|f(x_n)| \leq 10^{-2} \text{ és/vagy } |x_n - x^*| \leq 10^{-2}!$$

a) $f(x)=1-\ln(x+1)-x$ b) $f(x)=e^x-4x$

c) $f(x)=x^x-2$ d) $f(x)=\sin(x)+x-2$

e) $f(x)=x-e^{-x}$ f) $f(x)=xe^x-1$

4.1.14. Feladat. Határozza meg közelítőleg az $f(x)=0$ egyenletek x^* gyökét húrmódszerrel oly módon, hogy a közelítő megoldásként elfogadott x_n-re teljesüljenek az alábbiak:

$$|f(x_n)| \leq 10^{-3} \text{ és/vagy } |x_n - x^*| \leq 10^{-3}!$$

a) $f(x)=e^x+x-10$ b) $f(x)=1-x-\ln(1+x)$

c) $f(x)=x^3+x-1$ d) $f(x)=xe^x-1$

e) $f(x)=e^x-4x$ f) $f(x)=1-x-\sin(x)$

4.1.15. Feladat. Oldja meg a 4.1.13. feladat egyenleteit a 4.1.7. feladatban elkészített programmal!

4.1.16. Feladat. Oldja meg a 4.1.14. feladat egyenleteit a 4.1.8. feladatban elkészített programmal!

4.2. Érintőmódszer, szelőmódszer

4.2.1. Feladat. Adja meg a 4.1.1. feladatban szereplő egyenletek gyökeit az érintőmódszer segítségével $\epsilon = 10^{-3}$-os abszolút hibakorláttal!

4.2.2. Feladat. Adja meg a 4.1.1. feladatban szereplő egyenletek gyökeit a szelőmódszer segítségével $\epsilon = 10^{-3}$-os abszolút hibakorláttal!

4.2.3. Feladat. Bizonyítsa be, hogy az érintőmódszer lokálisan kvadratikusan konvergál a 4.1.1.d., a 4.1.1.e. és a 4.1.1.f. feladatok esetében!

4.2.4. Feladat. Bizonyítsa be, hogy az érintőmódszer az $x^3 - 3x^2 + 4 = 0$ egyenlet kisebbik gyökéhez másod, nagyobbik gyökéhez csak első rendben konvergál!

4.2.5. Feladat. Tekintse az $x_{n+1} = \frac{1}{2}\left(x_n + \frac{a}{x_n}\right)$, $a > 0$ úgynevezett Héron-sorozatot. Mutassa meg, hogy a sorozat megegyezik az $x^2 - a = 0$ egyenletre alkalmazott érintőmódszer által kapott sorozattal. Adja meg az x_0 kezdőpontok azon halmazát, amelyekből indítva konvergens lesz a sorozat!

4.2.6. Feladat. Általánosítsa a Héron-sorozatot az $x^k - a = 0$, $k > 0$ egyenletre! Mi mondható a konvergenciát adó x_0 kezdőpontok halmazáról?

4.2.7. Feladat. Bizonyítsa be, hogy a szelőmódszer a 4.1.1.a., a 4.1.1.b. és a 4.1.1.c. feladatok esetén $\frac{1+\sqrt{5}}{2}$ rendben lokálisan konvergál a gyök(ök)höz!

4.2.8. Feladat. Készítsen (Scilab) eljárást az érintőmódszer előállítására!
Az eljárás paraméterezése `Newton(f,df,a,t)` alakú legyen, ahol

`Newton`	az eljárás neve,
`f`	a függvény, aminek a gyökét keressük,
`df`	a függvény deriváltfüggvénye,
`a`	a kezdőpont,
`t`	a megoldás pontossága.

4.2.9. Feladat. Készítsen (Scilab) eljárást a szelőmódszer előállítására!
Az eljárás paraméterezése `Szelo(f,a,b,t)` alakú legyen, ahol

`Szelo`	az eljárás neve,
`f`	a függvény, aminek a gyökét keressük,
`a, b`	a két kezdőpont,
`t`	a megoldás pontossága.

4.2.10. Feladat. Oldja meg a 4.1.1. feladat egyenleteit a 4.2.8. feladatban elkészített program segítségével!

4.2.11. Feladat. Oldja meg a 4.1.1. feladat egyenleteit a 4.2.9. feladatban elkészített program segítségével!

4.2.12. Feladat. Határozza meg közelítőleg az $f(x)=0$ egyenletek x^* gyökét érintőmódszerrel oly módon, hogy a közelítő megoldásként elfogadott x_n-re teljesüljenek az alábbiak:

$$|f(x_n)| \leq 10^{-4} \text{ és/vagy } |x_n - x^*| \leq 10^{-4}!$$

a) $f(x) = e^x - 4x$ b) $f(x) = x^x - 2$
c) $f(x) = e^x + x - 10$ d) $f(x) = e^{2x} - \sin(x) - 2$
e) $f(x) = x^3 + x - 1$ f) $f(x) = 1 - x - \sin(x)$

4.2.13. Feladat. Határozza meg közelítőleg az $f(x)=0$ egyenletek x^* gyökét szelőmódszerrel oly módon, hogy a közelítő megoldásként elfogadott x_n-re teljesüljenek az alábbiak:

$$|f(x_n)| \leq 10^{-3} \text{ és/vagy } |x_n - x^*| \leq 10^{-3}!$$

a) $f(x) = x^3 + x - 1$ b) $f(x) = xe^x - 1$
c) $f(x) = \sin(x) + x - 2$ d) $f(x) = 1 + x + \sin(x)$
e) $f(x) = e^x - 6x$ f) $f(x) = e^{-x} + 5x$

4.2.14. Feladat. Oldja meg a 4.2.12. feladat egyenleteit a 4.2.8. feladatban elkészített program segítségével!

4.2.15. Feladat. Oldja meg a 4.2.13. feladat egyenleteit a 4.2.9. feladatban elkészített program segítségével!

4.2.16. Feladat. Vizsgálja meg, az elkészített programok segítségével, hogy a 4.1.1. feladat egyenleteinek megoldásában melyik módszer bizonyul hatékonyabbnak a szükséges időigény szempontjából!

4.3. Fixpont iteráció

4.3.1. Feladat. Mutassa meg, hogy az alábbi $g(x)$ függvények esetében az $x = g(x)$ fixpont egyenleteknek egyértelmű fixpontja van!

a) $g(x) = e^{-x}$ b) $g(x) = \frac{1}{2}\cos\left(\frac{1}{4}x\right)$
c) $g(x) = 2 - \ln(x)$ d) $g(x) = -e^x$
e) $g(x) = 1 - \ln(x+1)$ f) $g(x) = 1 - x^3$

4.3.2. Feladat. Határozza meg a 4.3.1. feladatbeli fixpont egyenletek fixpontját $\epsilon = 10^{-2}$ abszolút hibakorláttal az $x_{n+1} = g(x_n)$ fixpont iteráció segítségével alkalmasan választott x_0-ból kiindulva!

4.3.3. Feladat. Az alábbi egyenleteket hozza fixpont iterációra alkalmas $x=g(x)$ alakra, majd határozza meg a gyököket $\epsilon = 10^{-2}$ abszolút hibakorláttal!

a) $x^3 + x + 1 = 0$ b) $x + \ln(x) = 0$
c) $2x - \cos(x) = 0$

4.3.4. Feladat. Határozza meg az alábbi egyenletek gyökeit alkalmasan választott fixpont iteráció segítségével! Ha szükséges, használjon különböző átalakításokat a különböző gyökök esetében!

a) $e^x - 5x = 0$ b) $e^x - 2x - 2 = 0$

c) $x^3 - 5x^2 + 6 = 0$

4.3.5. Feladat. Hasonlítsa össze az $1 - x - \sin(x) = 0$ egyenlet fixpont iterációval történő megoldásainak konvergenciáját az alábbi $x = g(x)$ alakba való átírások esetén! Mi magyarázza a megfigyelhető konvergencia-sebesség eltéréseket? Az iterációt minden esetben az $x_0 = 0$ kezdőpontból indítsa!

a) $x = 1 - \sin(x)$ b) $x = \dfrac{1 + 3x - \sin(x)}{4}$

c) $x = \dfrac{1 + x - \sin(x)}{2}$

4.3.6. Feladat. Fixpont iteráció segítségével adja meg az alábbi fixpont egyenletek x^* gyökét közelítőleg úgy, hogy a közelítő megoldásként elfogadott x_n-re teljesüljön, $|x_n - x^*| \leq 10^{-3}$!

a) $x = \dfrac{e^x}{6}$ b) $x = \dfrac{-1 - \sin(x) + x}{2}$

c) $x = \ln(10 - x)$

4.3.7. Feladat. Készítsen (Scilab) eljárást a fixpont iteráció előállítására!
Az eljárás paraméterezése `Fixpont(f,a,t)` alakú legyen, ahol

`Fixpont`	az eljárás neve,
`f`	a függvény, aminek a fixpontját keressük,
`a`	a kezdőpont,
`t`	a megoldás pontossága.

4.3.8. Feladat. Oldja meg a 4.3.1. és a 4.3.6. feladatbeli fixpont egyenleteket a 4.3.7. feladatbeli program segítségével!

4.3.9. Feladat. Adja meg azokat a fixpont egyenleteket, amiknek a megoldása az alábbi x érték!

a) $x = \sqrt{2 + \sqrt{2 + \sqrt{2 + \ldots}}}$ b) $x = \sqrt{2 \cdot \sqrt{2 \cdot \sqrt{2 \cdot \ldots}}}$

c) $x = \cfrac{1}{2 + \cfrac{1}{2 + \cfrac{1}{2 + \ldots}}}$

4.3.10. Feladat. Mutassa meg, hogy az alábbi fixpont egyenleteknek 2, 2 illetve 1 fixpontja van!

a) $x = \sqrt{2 \cdot x}$ b) $x = \dfrac{1}{2 + x}$ c) $x = \sqrt{2 + x}$

4.3.11. Feladat. Határozza meg az alábbi fixpont iterációk esetében azokat az $x_0 \in \mathbb{R}$ pontoknak a halmazát, amelyek esetében x_n sorra a 4.3.9. feladatbeli x értékekhez konvergál!

a) $x_{n+1} = \sqrt{2+x_n}$ b) $x_{n+1} = \sqrt{2 \cdot x_n}$ c) $x_{n+1} = \dfrac{1}{2+x_n}$

5. fejezet

Numerikus integrálás

5.1. Newton-Cotes formulák

5.1.1. Feladat. Mutassa meg, hogy ha az $\int_a^b f(x)\,dx$ integrál közelítésére használt $\sum_{i=0}^n a_i f(x_i)$ kvadratúra formula pontos bármely konstans függvényre, akkor $\sum_{i=0}^n a_i = b-a$!

5.1.2. Feladat. Legyenek $x_0, x_1, \ldots, x_n$ különböző, rögzített $[a,b]$-beli számok. Mutassa meg, hogy az alábbi, a_i-kre felírt egyenletrendszernek mindig létezik egyértelmű megoldása!

$$\int_a^b x^j\,dx = \sum_{i=0}^n a_i x_i^j, \quad j = 0, 1, \ldots, n$$

5.1.3. Feladat. Mutassa meg, hogy az 5.1.2. feladat egyenletrendszerének megoldását adó a_i súlyokra igaz, hogy $\int_a^b p(x)\,dx = \sum_{i=0}^n a_i p(x_i)$ tetszőleges, legfeljebb n-edfokú p polinom esetén!

5.1.4. Feladat. Mutassa meg, hogy $n=0$ és $x_0 = \dfrac{a+b}{2}$ esetén a középpont szabályt kapja, ha megoldja az 5.1.2. feladatbeli egyenletrendszert!

5.1.5. Feladat. Mutassa meg, hogy $n=1$ és $x_0 = a$, $x_1 = b$ esetén a trapézszabályt kapja, ha megoldja az 5.1.2. feladatbeli egyenletrendszert!

5.1.6. Feladat. Mutassa meg, hogy $n=2$ és $x_0=a$, $x_1=\dfrac{a+b}{2}$, $x_2=b$ esetén a Simpson-szabályt kapja, ha megoldja az 5.1.2. feladatbeli egyenletrendszert!

5.1.7. Feladat. Mutassa meg, hogy alkalmasan megválaszthatók úgy az a_i súlyok és x_i osztópontok ($i=0,1,\ldots,n$), hogy az $\int_a^b p(x)\,dx=\sum_{i=0}^n a_i p(x_i)$ egyenlőség igaz legyen tetszőleges, legfeljebb $2n+1$-edfokú p polinom esetén!

5.1.8. Feladat. Adja meg az előző feladatbeli a_i súlyokat és x_i osztópontokat $n=1$ esetén, ha $[a,b]=[-1,1]$!

5.1.9. Feladat. Mutassa meg, hogy ha a középpont szabályt alkalmazza egy $f \in C^1$ függvényre, akkor a kapott érték megegyezik az f függvény $\dfrac{a+b}{2}$ pontbeli érintője, az $x=a$, az $x=b$ és az $y=0$ egyenesek által határolt előjeles területtel!

5.1.10. Feladat. Mutassa meg, hogy a Simpson-szabály a középpont szabály és a trapézszabály konvex kombinációja!

5.1.11. Feladat. Mutassa meg, hogy tetszőleges $[a,b]$ intervallumon konkáv, nemnegatív, folytonos függvényre alkalmazva akár az elemi, akár az összetett trapézszabályt, a kapott numerikus integrál értéke sosem lesz nagyobb, mint az f integrálja $[a,b]$-n!

5.1.12. Feladat. Mutassa meg, hogy tetszőleges $[a,b]$ intervallumon konkáv, nemnegatív, folytonos függvényre alkalmazva akár az elemi, akár az összetett középpont szabályt, a kapott numerikus integrál értéke sosem lesz kisebb, mint az f integrálja $[a,b]$-n!

5.1.13. Feladat. Mit mondhat, ha az előző két feladatban konkáv helyett konvex függvényt tekint? Indokolja a válaszát!

5.2. Összetett formulák, hibabecslés

5.2.1. Feladat. Határozza meg közelítőleg az $\int_0^\pi \sin(x)\,dx$ integrál értékét az összetett középpont szabályt alkalmazva az $x_i = \frac{\pi}{8} + \frac{\pi}{4} i$, $i = 0,1,2,3$ osztópontokat felhasználva! Adjon felső becslést az elkövetett hibára! Hány ekvidisztáns osztópontot kellene használnia, ha a hibát 10^{-2} alá szeretné szorítani? Számítsa ki ebben az esetben is az integrál közelítő értékét!

5.2.2. Feladat. Számítsa ki az alábbi integrálokat legfeljebb 10^{-2}-os hibával, ekvidisztáns osztópontokon alapuló összetett középpont szabályt alkalmazva!

a) $\displaystyle\int_0^1 x e^x\,dx$ b) $\displaystyle\int_{-1}^0 \cos(e^x)\,dx$ c) $\displaystyle\int_{-\frac{1}{2}}^{\frac{1}{2}} e^{x^2}\,dx$

d) $\displaystyle\int_0^\pi \cos\left(\frac{1}{2}x\right)\,dx$ e) $\displaystyle\int_1^2 \ln\left(\sqrt[3]{x}\right)\,dx$ f) $\displaystyle\int_0^1 \operatorname{arctg}(x)\,dx$

5.2.3. Feladat. Számítsa ki az alábbi integrálok közelítő értékét az összetett középpont szabályt alkalmazva $h = 0.5$, 0.25 és 0.125 lépésközzel! Becsülje meg az elkövetett hibát!

a) $\displaystyle\int_0^1 e^{\sin(x)}\,dx$ b) $\displaystyle\int_{-1}^0 \frac{x}{e^x}\,dx$ c) $\displaystyle\int_1^2 \ln\left(\sin(x)\right)\,dx$

5.2.4. Feladat. Készítsen (Scilab) eljárást az összetett középpont szabály előállítására!
Az eljárás paraméterezése `Osszkp(f,a,b,n)` alakú legyen, ahol

`Osszkp`	az eljárás neve,
`f`	a függvény, aminek az integrálját keressük,
`a`	az integrálás alsó határa,
`b`	az integrálás felső határa,
`n`	a részintervallumok száma.

5.2.5. Feladat. Számítsa ki az 5.2.2. és 5.2.3. feladatbeli integrálokat közelítőleg az 5.2.4. feladatban elkészített program segítségével $n = 10$, 20 és 40 darab egyenlő lépésközt alkalmazva!

5.2.6. Feladat. Határozza meg közelítőleg az $\int_0^\pi \sin(x)\,dx$ integrál értékét az összetett trapézszabályt alkalmazva az $x_i = \frac{\pi}{4} i$, $i = 0, 1, \ldots, 4$ osztópontokat felhasználva! Adjon felső becslést az elkövetett hibára! Hány ekvidisztáns osztópontot kellene használnia, ha a hibát 10^{-2} alá szeretné szorítani? Számítsa ki ebben az esetben is az integrál közelítő értékét!

5.2.7. Feladat. Számítsa ki az alábbi integrálokat legfeljebb 10^{-2}-os hibával, ekvidisztáns osztópontokon alapuló összetett trapézszabályt alkalmazva!

a) $\displaystyle\int_0^1 (x-1)e^x\,dx$ b) $\displaystyle\int_{-1}^0 \sin(e^{-x})\,dx$ c) $\displaystyle\int_{-1}^1 e^{|x|}\,dx$

d) $\displaystyle\int_{\frac{\pi}{2}}^{\pi} \sin\left(-\frac{1}{2}x\right)\,dx$ e) $\displaystyle\int_1^2 \ln\left(\sqrt[4]{x}\right)\,dx$ f) $\displaystyle\int_{-1}^0 \operatorname{arctg}(x)\,dx$

5.2.8. Feladat. Számítsa ki az alábbi integrálok közelítő értékét az összetett trapézszabályt alkalmazva $h = 0.5$, 0.25 és 0.125 lépésközzel! Becsülje meg az elkövetett hibát!

a) $\displaystyle\int_0^1 e^{\sin(x)}\,dx$ b) $\displaystyle\int_{-1}^0 xe^{-x}\,dx$ c) $\displaystyle\int_0^1 \ln\left(\cos(x)\right)\,dx$

5.2.9. Feladat. Készítsen (Scilab) eljárást az összetett trapézszabály előállítására!
Az eljárás paraméterezése `Trapez(f,a,b,n)` alakú legyen, ahol

`Trapez`	az eljárás neve,
`f`	a függvény, aminek az integrálját keressük,
`a`	az integrálás alsó határa,
`b`	az integrálás felső határa,
`n`	a részintervallumok száma.

5.2.10. Feladat. Számítsa ki az 5.2.7. és 5.2.8. feladatbeli integrálokat közelítőleg az 5.2.8. feladatban elkészített program segítségével $n = 20$, 40 és 80 darab egyenlő lépésközt alkalmazva!

5.2.11. Feladat. Határozza meg közelítőleg az $\int_0^\pi \sin(x)\,dx$ integrál értékét az összetett Simpson-szabályt alkalmazva az $x_i = \frac{\pi}{4} i$, $i = 0, 1, \ldots, 4$ osztópontokat felhasználva! Adjon felső becslést az elkövetett hibára! Hány ekvidisztáns osztópontot kellene használnia, ha a hibát 10^{-4} alá szeretné szorítani? Számítsa ki ebben az esetben is az integrál közelítő értékét!

5.2.12. Feladat. Számítsa ki az alábbi integrálokat legfeljebb 10^{-3}-os hibával, ekvidisztáns osztópontokon alapuló összetett Simpson-szabályt alkalmazva!

a) $\displaystyle\int_0^1 \operatorname{ch}(x)\,dx$ b) $\displaystyle\int_{-1}^1 e^{x^2}\,dx$ c) $\displaystyle\int_1^2 \ln\left(\sqrt{x}\right)\,dx$

d) $\displaystyle\int_{-1}^0 xe^{-x}\,dx$ e) $\displaystyle\int_{\frac{\pi}{2}}^{\pi} \cos(x)\,dx$ f) $\displaystyle\int_0^\pi \sin^2(x)\,dx$

5.2.13. Feladat. Számítsa ki az alábbi integrálok közelítő értékét az összetett Simpson-szabályt alkalmazva $h = 0.5$, 0.25 és 0.125 lépésközzel! Becsülje meg az elkövetett hibát!

a) $\displaystyle\int_{-1}^{0} xe^x\,dx$ b) $\displaystyle\int_{0}^{1} x^4\,dx$ c) $\displaystyle\int_{-1}^{0} \ln(2-x)\,dx$

5.2.14. Feladat. Készítsen (Scilab) eljárást az összetett Simpson-szabály előállítására!
Az eljárás paraméterezése `Simpson(f,a,b,n)` alakú legyen, ahol

`Simpson`	az eljárás neve,
`f`	a függvény, aminek az integrálját keressük,
`a`	az integrálás alsó határa,
`b`	az integrálás felső határa,
`n`	a részintervallumok száma.

5.2.15. Feladat. Számítsa ki az 5.2.12. és 5.2.13. feladatbeli integrálokat közelítőleg az 5.2.14. feladatban elkészített program segítségével $n = 10$, 20 és 40 darab egyenlő lépésközt alkalmazva!

5.2.16. Feladat. Bizonyítsa be, hogy ha f $[a,b]$-n Riemann integrálható függvény, akkor az összetett középpont szabályt alkalmazva az $x_i = \dfrac{b-a}{n}\left(i+\dfrac{1}{2}\right)$, $i = 0,1,\ldots,n-1$ osztópontokra tetszőleges pontossággal meghatározható az $\int_a^b f(x)\,dx$ integrál, ha $n \to \infty$!

5.2.17. Feladat. Bizonyítsa be, hogy ha f $[a,b]$-n Riemann integrálható függvény, akkor az összetett trapézszabályt alkalmazva az $x_i = a + \dfrac{b-a}{n} i$, $i = 0,1,\ldots,n$ osztópontokra tetszőleges pontossággal meghatározható az $\int_a^b f(x)$ integrál, ha $n \to \infty$!

5.2.18. Feladat. Bizonyítsa be, hogy ha f $[a,b]$-n Riemann integrálható függvény, akkor az összetett Simpson-szabályt alkalmazva az $x_i = a + \dfrac{b-a}{2n} i$, $i = 0,1,\ldots,2n$ osztópontokra tetszőleges pontossággal meghatározható az $\int_a^b f(x)\,dx$ integrál, ha $n \to \infty$!

5.2.19. Feladat. Mutassa meg, hogy az összetett középpont szabály segítségével tetszőleges pontossággal közelíthető a π értéke (csak racionális számok használatával)!

5.2.20. Feladat. Mutassa meg, hogy az összetett trapézszabály segítségével tetszőleges pontossággal közelíthető a π értéke (csak racionális számok használatával)!

5.2.21. Feladat. Mutassa meg, hogy az összetett Simpson-szabály segítségével tetszőleges pontossággal közelíthető a π értéke (csak racionális számok használatával)!

5.2.22. Feladat. Határozza meg π értékét 10^{-2} pontossággal az összetett középpont szabályt alkalmazva (csak racionális számok felhasználásával)!

5.2.23. Feladat. Határozza meg π értékét 10^{-6} pontossággal az összetett Simpson-szabályt alkalmazva (csak racionális számok felhasználásával)!

5.2.24. Feladat. Határozza meg e értékét 10^{-3} pontossággal az összetett Simpson-szabályt alkalmazva (csak racionális számok felhasználásával)!

5.2.25. Feladat. Határozza meg a $f(x)=\cos(x)$ függvény grafikonjának a $[0,\pi]$ feletti megforgatásával keletkező forgástest térfogatát 10^{-2} -os pontossággal!

5.2.26. Feladat. Határozza meg az $f(x)=e^{|x|}$ és a $g(x)=\sqrt{\cos(x)}+1$ függvények által bezárt véges síkidom területét 10^{-2}-os pontossággal!

5.2.27. Feladat. Oldja meg az alábbi egyenletet!

$$f(x)=\int_0^x e^{t^2}\,dt+x-1=0$$

6. fejezet

Szélsőérték feladatok

6.1. Aranymetszés szerinti keresés

6.1.1. Feladat. Adja meg a következő függvények minimumhelyét az aranymetszés szerinti keresés módszerével $\epsilon = 10^{-1}$-os abszolút hibakorláttal!

a) $f(x) = x^2 - 2x + 4,\ x \in [0,3]$

b) $f(x) = \cos(x) + 1,\ x \in \left[0, \dfrac{3\pi}{2}\right]$

c) $f(x) = e^{x^2} - 1,\ x \in [-1,2]$

d) $f(x) = e^{\sin^2(x)} - 1,\ x \in [-1,1]$

e) $f(x) = x^3 - 6x^2,\ x \in [0,3]$

f) $f(x) = x^3 - 6x^2,\ x \in [0,5]$

6.1.2. Feladat. Mutassa meg, hogy a 6.1.1. feladatbeli függvények unimodálisak a feltüntetett intervallumban!

6.1.3. Feladat. Határozza meg, hogy hány lépést kell végrehajtani a 6.1.1. feladatbeli függvények minimumhelyének aranymetszés szerinti keresés módszerével való közelítő meghatározásához, ha $\epsilon = 10^{-8}$-os abszolút hibakorlátot használunk és az utolsó intervallum középpontját fogadjuk el közelítésként!

6.1.4. Feladat. Alkalmazza az aranymetszés szerinti keresés módszerét a 6.1.1.c. és 6.1.1.d. feladatbeli függvényekre úgy, hogy a megállási kritérium az, hogy a függvényérték az utolsó intervallum középpontjában kisebb legyen, mint 10^{-3}!

6.1.5. Feladat. Alkalmazza az aranymetszés szerinti keresés módszerét a 6.1.1. feladatbeli függvények maximumhelyének közelítő meghatározására az adott intervallumon! Szükség esetén bontsa több részre az intervallumot!

6.1.6. Feladat. Készítsen (Scilab) eljárást az aranymetszés szerinti keresés módszerének előállítására!
Az eljárás paraméterezése `Aranymetszes(f,a,b,t)` alakú legyen, ahol

`Aranymetszes`	az eljárás neve,
`f`	a függvény, aminek a minimumhelyét keressük,
`a`	a kezdeti intervallum bal oldali végpontja,
`b`	a kezdeti intervallum jobb oldali végpontja,
`t`	a megoldás pontossága.

6.1.7. Feladat. Oldja meg a 6.1.1. feladatot a 6.1.6. feladatban elkészített programmal!

6.1.8. Feladat. Oldja meg a 6.1.4. feladatot a 6.1.6. feladatban elkészített programmal!

6.1.9. Feladat. Határozza meg annak a négyzet alapú hasábnak az éleit, amelynek felülete $24m^2$ és térfogata maximális!

6.2. Szimplex módszer

6.2.1. Feladat. Adja meg a 6.1.1. feladatbeli függvények minimumhelyét a szimplex módszerrel $\epsilon = 10^{-1}$-os abszolút hibakorláttal! Az induló szimplex minden esetben a $[0,1]$ intervallum legyen!

6.2.2. Feladat. Vizsgálja meg mi történik, ha a 6.1.1. feladat függvényei esetében más intervallumo(ka)t választunk kezdetben!

6.2.3. Feladat. Adja meg az alábbi függvények minimumhelyét a szimplex módszerrel $\epsilon =$ $= 10^{-1}$-os abszolút hibakorláttal (végtelen normában)! Az induló szimplex minden esetben legyen az egységvektorok és az azok összegvektora által meghatározott szimplex!

a) $f(x,y) = x^2 + y^2$ b) $f(x,y) = e^{x^2+y^2}$

c) $f(x,y) = e^{|x|+|y|}$ d) $f(x,y) = 2(x^2-2y)^2 + (x-1)^2$

e) $f(x,y,z) = 2\left(x+\frac{1}{2}\right)^2 + \left(y-\frac{1}{2}\right)^2 + (z-1)^2$ f) $f(x,y,z) = (x-y)^2 + (x-1)^2 + z^2$

6.2.4. Feladat. Mutassa meg, hogy a 6.2.3. feladatbeli függvényeknek egyértelmű globális minimumuk van!

6.2.5. Feladat. Készítsen (Scilab) eljárást a szimplex módszer előállítására!
Az eljárás paraméterezése `Szimplex(f,A,m)` alakú legyen, ahol

`Szimplex`	az eljárás neve,
`f`	a függvény, aminek a minimumhelyét keressük,
`A`	a kezdeti pontok mátrixa,
`m`	a lépésszám.

6.2.6. Feladat. Oldja meg a 6.2.1. feladatot a 6.2.5. feladatban elkészített programmal!

6.2.7. Feladat. Oldja meg a 6.2.3. feladatot a 6.2.5. feladatban elkészített programmal!

6.2.8. Feladat. Hasonlítsa össze futási idő szempontjából az aranymetszés szerinti keresést és a szimplex módszert a 6.1.1. feladatbeli függvények esetében!

6.2.9. Feladat. Keresse meg az alábbi függvény összes minimumhelyét a 6.2.5. feladatban elkészített programmal!

$$f(x,y) = x^4 - 2x^2 + y^5 - 5y.$$

6.3. Gradiens módszerek

6.3.1. Feladat. Adja meg az alábbi függvények minimumhelyét közelítőleg az állandó lépésközű, egyszerű gradiens módszerrel! Legyen a kezdővektor minden esetben az $(x_0, y_0) = (0.1, 0.2)$ vektor és a lépésköz 0.05 és számítsa ki az (x_3, y_3) vektort!

a) $f(x,y) = x^2 + xy + y^2$

b) $f(x,y) = |x| + |y|$

c) $f(x,y) = (2x + y^2)^2 + (y+1)^2$

6.3.2. Feladat. Mutassa meg, hogy a 6.3.1 feladatbeli függvényeknek egyértelmű globális minimumhelyük van!

6.3.3. Feladat. Készítsen (Scilab) eljárást az állandó lépésközű egyszerű gradiens módszer előállítására!
Az eljárás paraméterezése `Gradiens(f,df,x,h,m)` alakú legyen, ahol

`Gradiens`	az eljárás neve,
`f`	a függvény, amit minimalizálni kell,
`df`	az f gradiensfüggvénye,
`x`	a kezdővektor,
`h`	a lépésköz,
`m`	a lépésszám.

6.3.4. Feladat. Határozza meg a 6.2.3. és a 6.3.1. feladatbeli függvények minimumhelyét közelítőleg a 6.3.3. feladatbeli program segítségével különböző (0.1, 0.001 és 0.0001) lépésközök esetén, kiindulva az egységvektorok összegéből, mint kezdővektorból! Állítsa le az iterációt, ha az iterációszám meghaladja a 100-at, vagy az egymást követő értékek eltérése kisebb, mint 10^{-2}!

6.3.5. Feladat. Határozza meg az alábbi függvények minimumhelyét az optimális gradiens módszerrel $\epsilon = 10^{-1}$-es abszolút hibakorláttal! Legyen a kezdővektor minden esetben az $(x_0, y_0) = (1, 1)$ vektor!

a) $f(x,y) = (x+1)^2 + \left(y - \dfrac{1}{2}\right)^2$

b) $f(x,y) = e^{x^2+y^2}$

c) $f(x,y) = |x| + |y|$

d) $f(x,y) = 2(x^2 - 2y)^2 + (x+1)^2$

e) $f(x,y) = (x-y)^2 + (y+1)^2$

f) $f(x,y) = (2x+y+3)^2 + (x+2y+3)^2$

6.3.6. Feladat. Alkalmazza az optimális gradiens módszert az $f \in \mathbb{R}^n \to \mathbb{R}$, $f(\mathbf{x}) = \frac{1}{2}||A\mathbf{x} - \mathbf{b}||_2^2$ függvényre, ahol $\mathbf{A}$ szimmetrikus, pozitív definit mátrix! Mutassa meg, hogy a módszer tetszőleges $\mathbf{x}_0$ vektorból indítva konvergál a minimumhelyhez!

6.3.7. Feladat. Készítsen (Scilab) eljárást az optimális lépésközű gradiens módszer előállítására $f(\mathbf{x}) = \frac{1}{2}||A\mathbf{x} - \mathbf{b}||_2^2$ alakú függvények esetén!
Az eljárás paraméterezése `OptGrad(x,A,b,t)` alakú legyen, ahol

`OptGrad`	az eljárás neve,
`x`	a kezdővektor,
`A`	az együtthatómátrix,
`b`	az egyenletrendszer jobb oldala,
`t`	a megoldás pontossága.

6.3.8. Feladat. Alkalmazza a 6.3.7. feladatban elkészített programot az alábbi lineáris egyenletrendszerek megoldására!

a) $\begin{aligned} 2x + y &= -3 \\ x + 2y &= -3 \end{aligned}$ b) $\begin{aligned} 4x + y &= 4 \\ x + 5y &= 1 \end{aligned}$ c) $\begin{aligned} 3x + y + z &= 5 \\ x + 4y + 2z &= 7 \\ x + 2y + 8z &= 11 \end{aligned}$

6.3.9. Feladat. Határozza meg a 6.3.5. feladatbeli függvények minimumhelyét a konjugált gradiens módszerrel $\epsilon = 10^{-1}$-es abszolút hibakorláttal! Legyen a kezdővektor minden esetben az $(x_0, y_0) = (1,1)$ vektor! Hasonlítsa össze az eredményeket az optimális gradiens módszer alkalmazása során kapott eredményekkel!

6.3.10. Feladat. Alkalmazza a konjugált gradiens módszert az $f : \mathbb{R}^n \to \mathbb{R}$, $f(\mathbf{x}) = \frac{1}{2}||\mathbf{A}\mathbf{x} - \mathbf{b}||_2^2$ függvényre, ahol $\mathbf{A}$ szimmetrikus, pozitív definit mátrix! Mutassa meg, hogy a módszer tetszőleges $\mathbf{x}_0$ vektorból indítva konvergál a minimumhelyhez!

6.3.11. Feladat. Készítsen (Scilab) eljárást a konjugált gradiens módszer előállítására $f(\mathbf{x}) = \frac{1}{2}||A\mathbf{x} - \mathbf{b}||_2^2$ alakú függvények esetén!
Az eljárás paraméterezése `KonGrad(x,A,b,t)` alakú legyen, ahol

`KonGrad`	az eljárás neve,
`x`	a függvény, aminek a minimumhelyét keressük,
`A`	a kezdeti együtthatómátrix,
`b`	az egyenletrendszer jobb oldala,
`t`	a megoldás pontossága.

6.3.12. Feladat. Alkalmazza a 6.3.11. feladatban elkészített programot a 6.3.8. feladatbeli lineáris egyenletrendszerek megoldására! Hasonlítsa össze az eredményeket az optimális gradiens módszer alkalmazása során kapott eredményekkel!

6.3.13. Feladat. Mutassa meg, hogy az alábbi függvények Hesse-mátrixa pozitív definit és az $S(x_0, y_0) = \left\{(x, y) \in \mathbb{R}^2 | f(x, y) \le f(x_0, y_0)\right\}$ szinthalmazai konvexek és zártak!

a) $f(x,y)=x^2+y^2$

b) $f(x,y)=e^{x^2+y^2}$

c) $f(x,y)=(x-y)^2+(y+1)^2$

6.3.14. Feladat. Mutassa meg, hogy a 6.3.13. feladatbeli függvényekre konvergens a konjugált gradiens módszer!

6.3.15. Feladat. Készítsen (Scilab) eljárást az optimális gradiens módszer előállítására általános, differenciálható $f(x)$ függvény esetén!
Az eljárás paraméterezése `DOptGrad(f,df,x,m)` alakú legyen, ahol

`DOptGrad`	az eljárás neve,
`f`	a függvény, aminek a minimumhelyét keressük,
`df`	az f gradiensfüggvénye,
`x`	a kezdőpont,
`m`	a lépésszám.

6.3.16. Feladat. Készítsen (Scilab) eljárást a konjugált gradiens módszer előállítására általános, differenciálható $f(x)$ függvény esetén!
Az eljárás paraméterezése `DKonGrad(f,df,x,m)` alakú legyen, ahol

`DKonGrad`	az eljárás neve,
`f`	a függvény, aminek a minimumhelyét keressük,
`df`	az f gradiensfüggvénye,
`x`	a kezdőpont,
`m`	a lépésszám.

6.3.17. Feladat. Alkalmazza a 6.3.15. feladatban elkészített programot a 6.3.5. feladatok megoldására!

6.3.18. Feladat. Alkalmazza a 6.3.16. feladatban elkészített programot a 6.3.5. feladatok megoldására! Hasonlítsa össze az optimális gradiens módszerrel kapott eredményekkel!

7. fejezet

Ortogonális transzformációk és alkalmazásaik

7.1. Ortogonális transzformációk és ortogonális felbontások

7.1.1. Feladat. Adott $\mathbf{S} = \mathbf{u}\mathbf{v}^{\mathrm{T}} \in \mathbb{R}^{n\times n}$ 1 rangú mátrix és valamely $\mathbf{A} \in \mathbb{R}^{n\times n}$ mátrix, $\mathbf{a} \in \in \mathbb{R}^n$ vektor esetén hogyan határozható meg az $\alpha\mathbf{S}$, $\mathbf{S}\mathbf{a}$, $\mathbf{a}^T\mathbf{S}$, $\mathbf{S}\mathbf{A}$ és $\mathbf{A}\mathbf{S}$ kifejezések értéke? Mennyi az ehhez szükséges aritmetikai műveletek száma?

7.1.2. Feladat. Hogyan határozható meg egyszerűen (egy rangú módosítással) az

$$\mathbf{A} = \begin{pmatrix} 2 & 1 & 1 & \cdots & 1 \\ 1 & 2 & 1 & \cdots & 1 \\ 1 & 1 & 2 & \cdots & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & 1 & \cdots & 2 \end{pmatrix}$$

mátrix inverze?

7.1.3. Feladat. Igazolja, hogy tetszőleges $\mathbf{R} \in \mathbb{R}^{n\times n}$ nem-szinguláris felső trianguláris mátrix és tetszőleges $\mathbf{a} \in \mathbb{R}^n$ vektor esetében az $\mathbf{A} = \mathbf{R}^{\mathrm{T}}\mathbf{R} + \mathbf{a}\mathbf{a}^{\mathrm{T}}$ mátrix pozitív definit. Igaz-e hasonló állítás a $\mathbf{B} = \mathbf{R}^{\mathrm{T}}\mathbf{R} - \mathbf{a}\mathbf{a}^{\mathrm{T}}$ mátrixra? Milyen $\mathbf{a}$ vektorokra lesz ez is pozitív definit?

7.1.4. Feladat. Hány olyan 3×3-as ortogonális $\mathbf{Q}$ mátrix van, amelynek minden komponense 0, 1 vagy -1?

7.1.5. Feladat. Keressen példát olyan $n\times n$-es ortogonális $\mathbf{Q}$ mátrixra, amelynek *nem* minden sajátértéke valós.

7.1.6. Feladat. Lehet-e egy $\mathbf{S} \in \mathbb{R}^{n\times n}$, $\mathbf{S} \neq \mathbf{I}$ mátrix *egyszerre* elemi forgató (más néven Householder) és elemi tükröző (más néven Givens) mátrix?

7.1.7. Feladat. Határozza meg azokat a $\mathbf{H}_{AB}, \mathbf{H}_{BC}, \ldots, \mathbf{H}_{HA}$ elemi tükröző mátrixokat, amelyek az origóba helyezett $ABCDEFGH$ egységkocka csúcsait sorra a következő csúcsba viszik át. Mit állíthatunk a kapott mátrixok szorzatairól?

7.1.8. Feladat. Határozza meg azokat a $\mathbf{H}_{12}, \mathbf{H}_{23}, \dots, \mathbf{H}_{n1}$ elemi tükröző mátrixokat, amelyek az n dimenziós $\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_n$ egységvektorokat sorra a következő egységvektorba viszik át. Mit állíthatunk a kapott mátrixok szorzatairól?

7.1.9. Feladat. Igazolja, hogy bármely n dimenziós elemi tükrözőmátrixnak a $+1$ szám $(n-1)$-szeres, a -1 pedig egyszeres multiplicitású sajátértéke.

7.1.10. Feladat. Van-e értelme a 7.1.7. Feladatnak elemi forgató mátrixok esetében? Hogyan fogalmazná át az $ABCDEFGH$ egységkocka csúcsainak transzformálási követelményeit ebben az esetben?

7.1.11. Feladat. Határozza meg azokat a $\mathbf{G}_{12}, \mathbf{G}_{23}, \dots, \mathbf{G}_{n1}$ elemi forgató mátrixokat, amelyek az n dimenziós $\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_n$ egységvektorokat sorra a következő egységvektorba viszik át. Mit állíthatunk a kapott mátrixok szorzatairól?

7.1.12. Feladat. Határozza meg a következő reguláris mátrixok **QR** ortogonális trianguláris felbontását a Gram-Schmidt ortogonalizációs eljárás segítségével.

$$\mathbf{A} = \begin{pmatrix} 2 & -1 & 1 & -2 \\ 2 & -1 & 1 & 2 \\ 0 & 0 & -2 & 1 \\ 0 & -2 & 2 & 0 \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} -2 & 2 & -2 & 1 \\ 2 & 2 & -2 & -2 \\ 0 & -1 & -1 & 0 \\ 1 & 0 & 2 & 1 \end{pmatrix}, \quad \mathbf{C} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 3 & 0 & 1 \\ 1 & 0 & 1 & 3 \\ 1 & 0 & 0 & 3 \end{pmatrix}.$$

7.1.13. Feladat. Határozza meg a következő mátrixok **QR** ortogonális trianguláris felbontását elemi tükrözőmátrixok (Householder mátrixok) segítségével. Figyelje meg, hogy itt már *nem* minden **A** mátrix reguláris.

$$\mathbf{A} = \begin{pmatrix} 2 & -2 & 0 & -1 \\ 1 & 0 & 0 & 0 \\ 0 & -1 & 2 & -2 \\ -2 & -2 & 1 & -1 \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} 0 & 2 & -2 & 0 \\ 0 & -2 & 2 & -1 \\ -1 & 1 & 1 & 0 \\ -1 & 1 & 1 & 0 \end{pmatrix}, \quad \mathbf{C} = \begin{pmatrix} 0 & 2 & -2 & 0 \\ 0 & -2 & 2 & -1 \\ -1 & 1 & 1 & 0 \\ -1 & 1 & 1 & 2 \end{pmatrix}.$$

7.1.14. Feladat. Határozza meg a következő mátrixok **QR** ortogonális trianguláris felbontását elemi forgatómátrixok (Givens mátrixok) segítségével. Figyelje meg, hogy itt már *nem* minden **A** mátrix reguláris.

$$\mathbf{A} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 2 \\ 0 & 2 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} 0 & 1 & 0 & 2 \\ 2 & 2 & 0 & 2 \\ 0 & 0 & 2 & 2 \\ 0 & 1 & 0 & 2 \end{pmatrix}, \quad \mathbf{C} = \begin{pmatrix} 1 & 0 & -1 & 2 \\ 1 & 2 & 2 & -1 \\ 1 & 2 & 1 & 2 \\ -1 & 0 & 2 & 0 \end{pmatrix}.$$

7.1.15. Feladat. Bizonyítsa be az $\mathbf{A} = \mathbf{QR}$ ortogonális trianguláris felbontás felhasználásával, hogy az $\mathbf{A} \in \mathbb{R}^{n\times n}$ mátrix oszlopvektorait $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n$-nel jelölve teljesül az $|\det \mathbf{A}| \leq$ $\leq \prod_{j=1}^{n} \|\mathbf{a}_j\|_2$ egyenlőtlenség.

7.1.16. Feladat. Transzformálja elemi tükröző mátrixokkal végzett ortogonális hasonlósági transzformációkkal felső Hessenberg alakra az alábbi mátrixokat.

$$\mathbf{A} = \begin{pmatrix} -2 & -1 & 0 & 1 \\ -1 & 2 & 0 & 1 \\ -1 & -1 & -2 & 2 \\ 0 & 2 & -2 & -1 \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} 1 & 2 & -2 & -2 \\ 0 & 0 & -2 & 2 \\ 0 & -2 & -1 & 1 \\ -1 & 0 & 1 & -1 \end{pmatrix}, \quad \mathbf{C} = \begin{pmatrix} -1 & -2 & -2 & 0 \\ -1 & 0 & 2 & -2 \\ 0 & 0 & -2 & 0 \\ 2 & -1 & -2 & -1 \end{pmatrix}.$$

7.1.17. Feladat. Transzformálja elemi forgató mátrixokkal végzett ortogonális hasonlósági transzformációkkal felső Hessenberg alakra az előző példákban szereplő mátrixokat.

7.1.18. Feladat. Írjon elemi tükrözőmátrixok segítségével $\mathbf{A} = \mathbf{QR}$ ortogonális trianguláris felbontást végző eljárást.
Az eljárás paraméterezése `QRTukr(A)` alakú legyen, ahol
`QRTukr` az eljárás neve,
`A` a felbontandó $\mathbf{A}$ mátrix.

7.1.19. Feladat. Alkalmazza a 7.1.18. Feladat eljárását a fejezet példamátrixainak $\mathbf{A} = \mathbf{QR}$ felbontására.

7.1.20. Feladat. Írjon elemi forgatómátrixok segítségével $\mathbf{A} = \mathbf{QR}$ ortogonális trianguláris felbontást végző eljárást.
Az eljárás paraméterezése `QRForg(A)` alakú legyen, ahol
`QRForg` az eljárás neve,
`A` a felbontandó $\mathbf{A}$ mátrix.

7.1.21. Feladat. Alkalmazza a 7.1.20. Feladat eljárását a fejezet példamátrixainak $\mathbf{A} = \mathbf{QR}$ felbontására.

7.2. Általánosított inverz, SVD

7.2.1. Feladat. Határozza meg az alábbi $\mathbf{A}$ mátrix általánosított (más néven Moore–Penrose) inverzét:

$$\mathbf{A} = \begin{pmatrix} 1 & 0 & 1 & 1 \\ 1 & 0 & -1 & 0 \\ 1 & 1 & 0 & 1 \end{pmatrix}.$$

7.2.2. Feladat. Igazolja, hogy a mátrixok általánosított inverze *nem* folytonos függvénye a mátrixkomponenseknek. Keressen egyszerű ellenpéldát.

7.2.3. Feladat. Ez a feladat az (általánosított) QR felbontás és az általánosított inverz kapcsolatáról szól. Igazolja, hogy ha valamely $1 \le n \le m$ értékekre az $\mathbf{A} \in \mathbb{R}^{m\times n}$ oszlopreguláris mátrix felbontása

$$\mathbf{A} = \mathbf{Q}\begin{pmatrix} \mathbf{R} \\ \hline \mathbf{O} \end{pmatrix}$$

alakú, ahol $\mathbf{R} \in \mathbb{R}^{n\times n}$ felső trianguláris és $\operatorname{rang}(\mathbf{A}) = \operatorname{rang}(\mathbf{Q}) = n$, akkor $\mathbf{A}^+ = (\mathbf{R}^+ \mid \mathbf{O})\mathbf{Q}^+ =$
$= (\mathbf{R}^{-1} \mid \mathbf{O})\mathbf{Q}^{\mathrm{T}}$.

7.2.4. Feladat. Igazolja, hogy ha az $\mathbf{A} \in \mathbb{R}^{m\times n}$ mátrix oszlopreguláris és $m > n$, akkor az $\mathbf{Ax} = \mathbf{b}$ lineáris egyenletrendszer normálmegoldásának meghatározása ekvivalens az

$$\begin{pmatrix} \alpha\mathbf{I} & \mathbf{A} \\ \beta\mathbf{A}^{\mathrm{T}} & \mathbf{O} \end{pmatrix}\begin{pmatrix} \mathbf{r} \\ \mathbf{x} \end{pmatrix} = \begin{pmatrix} \mathbf{b} \\ \mathbf{0} \end{pmatrix}$$

$m+n$ ismeretlenes egyenletrendszer megoldásával, melyben α és β tetszőleges, nullától különböző valós paraméterek.

7.2.5. Feladat. Legyenek $\mathbf{A}, \mathbf{B}, \mathbf{C} \in \mathbb{R}^{m\times n}$ és $\mathbf{D} \in \mathbb{R}^{n\times m}$ tetszőleges mátrixok. Bizonyítsa be, hogy az

$$\begin{aligned} \mathbf{AX} \quad &- \mathbf{YB} = \mathbf{B} \\ \mathbf{XB}^{\mathrm{T}} \quad &- \mathbf{A}^{\mathrm{T}}\mathbf{Y} = \mathbf{D} \end{aligned}$$

mátrix-egyenletrendszer megoldásának unicitásához, azaz hogy *legfeljebb egy* $\mathbf{X}$, $\mathbf{Y}$ megoldása legyen, szükséges, hogy az $\mathbf{A}$ és a $\mathbf{B}$ mátrixok szinguláris értékei különbözők legyenek.

7.2.6. Feladat. Határozza meg a $\mathbf{H}_4$ Hilbert-mátrix SVD-felbontását.

7.2.7. Feladat. Igazolja, hogy ha az $\mathbf{A}$ mátrix SVD-felbontása $\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^{\mathrm{T}}$, akkor Frobenius-normája az $\|\mathbf{A}\|_F = \sqrt{\sum_{j=1}^{r} \sigma_j^2}$ képlettel fejezhető ki szinguláris értékeinek segítségével.

7.3. A sajátértékszámítás alapjai

7.3.1. Feladat. A $p_{\mathbf{A}}(\lambda)$ karakterisztikus polinom különböző felírásmódjait összehasonlítva igazolja a következő állításokat:

a) $\alpha_1 = \sum_{j=1}^{n} a_{jj} = \operatorname{tr}\mathbf{A} = \sum_{j=1}^{n} \lambda_j$,

b) $(-1)^{n+1}\alpha_n = \det\mathbf{A} = \prod_{j=1}^{n} \lambda_j$.

7.3.2. Feladat. Határozza meg az $\mathbf{a} = (1\,,1\,,1\,,1)^{\mathrm{T}}$ vektorhoz tartozó $\mathbf{A} = \mathbf{I} + \mathbf{a}\mathbf{a}^{\mathrm{T}} \in \mathbb{R}^{4\times 4}$ mátrix sajátértékeit és sajátvektorait.

7.3.3. Feladat. Hogyan általánosítható a 7.3.2. Feladat eredménye tetszőleges $n \geq 4$-re?

7.3.4. Feladat. Legyen $\mathbf{D}=\operatorname{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n) \in \mathbb{R}^{n\times n}$ tetszőleges valós diagonális mátrix, és $\mathbf{v} \in \mathbb{R}^n$ tetszőleges valós egységvektor: $\mathbf{v}^{\mathrm{T}}\mathbf{v} = 1$. Mutassa meg, hogy ekkor az $\mathbf{A} = (\mathbf{I} - 2\mathbf{v}\mathbf{v}^{\mathrm{T}})\mathbf{D}(\mathbf{I} - 2\mathbf{v}\mathbf{v}^{\mathrm{T}})$ mátrix szimmetrikus, sajátértékei a λ_k számok, továbbá a λ_k-hoz tartozó sajátvektora éppen az $\mathbf{I} - 2\mathbf{v}\mathbf{v}^{\mathrm{T}}$ mátrix k-dik oszlopa.

7.3.5. Feladat. Határozza meg az

$$\mathbf{A} = \begin{pmatrix} \beta & \alpha & & & & \\ \alpha & \beta & \alpha & & & \\ & \alpha & \beta & \alpha & & \\ & & \ddots & \ddots & \ddots & \\ & & & \alpha & \beta & \alpha \\ & & & & \alpha & \beta \end{pmatrix}$$

mátrix sajátértékeit és sajátvektorait.

7.3.6. Feladat. Mutassa meg, hogy az $\mathbf{A}^H = \mathbf{A}$ feltétel csak *elegendő* feltétele a sajátértékek valósságának, továbbá az állítás nem lenne igaz tetszőleges (komplex) $\mathbf{A}$ -ra, ha csak az $\mathbf{A}^T = \mathbf{A}$ feltételt követelnénk meg.

7.3.7. Feladat. A mátrixok $\mathbf{U}^{\mathrm{H}}\mathbf{A}\mathbf{U}=\mathbf{R}$ Schur-normálformájából kiindulva mutassa meg, hogy tetszőleges valós, szimmetrikus $\mathbf{A}$ mátrix összes sajátértéke valós.

7.3.8. Feladat. Mutassa meg, hogy ha az $\mathbf{A}$ és a $\mathbf{B}$ mátrixok sajátértékei megegyeznek, továbbá mindkét mátrix diagonalizálható, akkor hasonlóak.

7.3.9. Feladat. Tudjuk, hogy tetszőleges $\mathbf{U}$ (felső) trianguláris mátrix sajátértékei $\lambda_i = u_{ii}$ alakban rögtön felírhatók. Bizonyítsa be ennek általánosításaként, hogy tetszőleges $p(x)$ polinomot véve a $\mathbf{V}=p(\mathbf{U})$ mátrixpolinom μ_i sajátértékei pontosan a $\mu_i=p(\lambda_i)$ számok lesznek.

7.3.10. Feladat. A 7.3.9. Feladatot és a mátrixok Schur-féle normálformáját felhasználva igazolja, hogy tetszőleges $\mathbf{A}$ mátrix és tetszőleges $p(x)$ polinom esetén a $p(\mathbf{A})$ mátrixpolinom sajátértékei pontosan a $p(\lambda_i)$ értékek.

7.3.11. Feladat. Az $\mathbf{A} \in \mathbb{R}^{n\times n}$ mátrix λ_k sajátértékének *algebrai* multiplicitása m_k, ha λ_k a $p_{\mathbf{A}}(\lambda)$ karakterisztikus polinom m_k-szoros gyöke. A λ_k *geometriai* multiplicitása n_k, ha a λ_k-hoz tartozó sajátaltér dimenziója n_k, azaz legfeljebb n_k lineárisan független λ_k-hoz tartozó sajátvektor adható meg.
Igazolja, hogy a fenti számokra mindig teljesülnek az $1 \le n_k \le m_k \le n$ egyenlőtlenségek.

7.3.12. Feladat. Keressen példákat olyan $\mathbf{A} \in \mathbb{R}^{n\times n}$ mátrixokra, amelyek esetében a 7.3.11. Feladatban megadott egyenlőtlenségekben valahol (esetleg több helyen is) egyenlőség áll.
Tud-e olyan (elegendő) feltételeket mondani $\mathbf{A}$-ra, amelyekből valahol egyenlőség következik?

7.3.13. Feladat. Igazolja a mátrixok $\rho(\mathbf{A})$ spektrálsugarára vonatkozó ismert $\rho(\mathbf{A}) \le \|\mathbf{A}\|$ egyenlőtlenséghez kapcsolódó következő állítást.

Állítás. *Ha $\rho(\mathbf{A}) < 1$, akkor megadható olyan $\|.\|$ mátrixnorma, amelyben $\rho(\mathbf{A}) \le \|\mathbf{A}\| < 1$.*

7.3.14. Feladat. Igazolja, hogy bármely $\mathbf{A} \in \mathbb{R}^{n\times n}$ mátrix $\|.\|_2$ és $\|.\|_F$ Frobenius normájára

$$\|\mathbf{A}\|_2 \le \|\mathbf{A}\|_F \le \sqrt{n}\,\|\mathbf{A}\|_2 \,.$$

7.3.15. Feladat. Keressen példákat olyan $\mathbf{A}$ mátrixokra, melyekkel a 7.3.14. Feladat egyenlőtlenségeiben valahol egyenlőség teljesül.
Igazolja a következő állítást:

Állítás. *Az $\|\mathbf{A}\|_2 = \|\mathbf{A}\|_F$ egyenlőség akkor és csak akkor teljesül, ha $\mathbf{A} = \mathbf{a}\mathbf{b}^T$ alakban megadható 1 rangú mátrix.*

7.3.16. Feladat. Milyen kondíciószámokra vonatkozó becslést kap a 7.3.14. Feladat egyenlőtlenségeiből?

7.3.17. Feladat. Bizonyítsa be a Gersgorin-tétel felhasználásával, hogy ha az $\mathbf{A}$ mátrix szimmetrikus, diagonális domináns és diagonális elemei pozitívak, azaz $a_{jj}>0$ minden $1\le j\le n$-re, akkor $\mathbf{A}$ pozitív definit.

7.3.18. Feladat. A sajátértékekre vonatkozó becslések segítségével vizsgálja meg, hogy az

$$\mathbf{A}=\begin{pmatrix} \frac{1}{2} & \frac{1}{4} & \frac{1}{8} & \frac{1}{16} & \cdots & \frac{1}{2^n} \\ \frac{1}{4} & \frac{1}{2} & \frac{1}{8} & \frac{1}{16} & \cdots & \frac{1}{2^n} \\ \frac{1}{4} & \frac{1}{8} & \frac{1}{2} & \frac{1}{16} & \cdots & \frac{1}{2^n} \\ \frac{1}{4} & \frac{1}{8} & \frac{1}{16} & \frac{1}{2} & \cdots & \frac{1}{2^n} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \frac{1}{2^n} \\ \frac{1}{4} & \frac{1}{8} & \frac{1}{16} & \cdots & \frac{1}{2^n} & \frac{1}{2} \end{pmatrix}$$

mátrixnak lehet-e sajátértéke $\lambda = 0$, $\lambda = 1$ vagy $\lambda = \frac{1}{2}$.

7.3.19. Feladat. Bizonyítsa be, hogy tetszőleges $\mathbf{B} \in \mathbb{R}^{n\times n}$ mátrixra az $\mathbf{A} \in \mathbb{R}^{2n\times 2n}$,

$$\mathbf{A}=\begin{pmatrix} \mathbf{I} & \mathbf{B} \\ \mathbf{B}^{\mathrm{T}} & \mathbf{I} \end{pmatrix}$$

mátrix pontosan akkor lesz pozitív definit, ha $\rho(\mathbf{B}^{\mathrm{T}}\mathbf{B}) < 1$.

7.3.20. Feladat. Bizonyítsa be, hogy az ortogonális hasonlósági transzformációk megőrzik mind a $\|.\|_2$, mind a $\|.\|_F$ normára vonatkozó kondíciószámot, tehát ha $\mathbf{Q}, \mathbf{A} \in \mathbb{R}^{n\times n}$ és $\mathbf{Q}$ ortogonális, akkor $\mathrm{cond}(\mathbf{A}) = \mathrm{cond}(\mathbf{Q}^{\mathrm{T}}\mathbf{A}\mathbf{Q})$.

7.4. Sajátérték-számítás numerikus módszerekkel

7.4.1. Feladat. Közelítse az alábbi mátrix sajátértékeit LR-transzformációval:

$$\mathbf{A}=\begin{pmatrix} -42 & -70 & 92 & 92 \\ 19 & 31 & -38 & -38 \\ -65 & -109 & 146 & 146 \\ 57 & 93 & -126 & -126 \end{pmatrix}.$$

7.4.2. Feladat. Közelítse az alábbi mátrix sajátértékeit LR-transzformációval:

$$\mathbf{A}=\begin{pmatrix} 15 & 12 & -18 & 12 \\ -6 & -3 & 1 & 5 \\ 12 & 12 & -15 & 12 \\ 12 & 12 & -7 & 4 \end{pmatrix}.$$

7.4.3. Feladat. Közelítse az alábbi mátrix sajátértékeit LR-transzformációval:

$$\mathbf{A}=\begin{pmatrix} 26 & 32 & 46 & -25 & 22 \\ 128 & 162 & 218 & -117 & 126 \\ -153 & -197 & -263 & 140 & -152 \\ -134 & -176 & -230 & 121 & -138 \\ -59 & -77 & -101 & 54 & -62 \end{pmatrix}.$$

7.4.4. Feladat. Mivel az LR-transzformáció a nemlétező $\mathbf{L}_k\mathbf{R}_k$ trianguláris felbontás miatt elakadhat a k-dik lépésben, próbálja meg helyette tetszőleges reguláris $\mathbf{A}$ sajátértékeinek közelítésére a mátrixok általánosított trianguláris felbontását használó következő *általánosított LR-transzformációt* alkalmazni amely nem akadhat el.
Legyen $\mathbf{A}_1 = \mathbf{A}$, és adott $\mathbf{P}_k\mathbf{A}_k = \mathbf{L}_k\mathbf{R}_k$-ra $\mathbf{A}_{k+1} = \mathbf{R}_k\mathbf{P}_k^{\mathrm{T}}\mathbf{L}_k$.
Próbálja ki az $\mathbf{A} = \begin{pmatrix} 1 & 3 \\ 2 & 0 \end{pmatrix}$ mátrixra az algoritmust.

7.4.5. Feladat. Mutassa meg, hogy az alábbi $\mathbf{A}$ mátrixra az LR-transzformáció nem konvergál, de a QR-transzformáció igen.

$$\mathbf{A} = \begin{pmatrix} 26 & 32 & 46 & -25 & 22 \\ 128 & 162 & 218 & -117 & 126 \\ -153 & -197 & -263 & 140 & -152 \\ -134 & -176 & -230 & 121 & -138 \\ -59 & -77 & -101 & 54 & -62 \end{pmatrix}.$$

7.4.6. Feladat. Mutassa meg, hogy az alábbi $\mathbf{A}$ mátrixra az LR-transzformáció nem konvergál, de a QR-transzformáció igen.

$$\mathbf{A} = \begin{pmatrix} -2 & 0 & 0 & 3 & 0 \\ 11 & -2 & 11 & 20 & -22 \\ 11 & 0 & 9 & 20 & -22 \\ 0 & 0 & 0 & -8 & 0 \\ 0 & 0 & 0 & 3 & -2 \end{pmatrix}.$$

7.4.7. Feladat. Közelítse az alábbi $\mathbf{A}$ mátrix sajátértékeit. Alkalmazható-e az LR-transzformáció? Mit eredményez a QR-transzformáció?

$$\mathbf{A} = \begin{pmatrix} 0 & 3 & -5 \\ 4 & -2 & 2 \\ -4 & 3 & -1 \end{pmatrix}.$$

7.4.8. Feladat. Ismert eredmény, hogy ha a QR transzformáció $\mathbf{A}_1$ kiinduló mártixa felső Hessenberg alakú, akkor az összes többi $\mathbf{A}_k$ is felső Hessenberg alakú lesz. A transzformáció k-dik lépése végrehajtható az $\mathbf{S}_{21}^k, \mathbf{S}_{32}^k, \ldots, \mathbf{S}_{n,n-1}^k$ elemi forgató mátrixok segítségével, melyek az $\mathbf{A}_k$ mátrix főátló alatti elemeit nullázzák ki:

$$\mathbf{A}_{k+1} = \mathbf{S}_{n,n-1}^k \cdots \mathbf{S}_{32}^k\mathbf{S}_{21}^k\mathbf{A}_k(\mathbf{S}_{21}^k)^{\mathrm{T}}(\mathbf{S}_{32}^k)^{\mathrm{T}} \cdots (\mathbf{S}_{n,n-1}^k)^{\mathrm{T}}.$$

Mutassa meg, hogy ezekkel a képletekkel $O(n^2)$ művelettel „helyben” kiszámolható az $\mathbf{A}_{k+1}$ mátrix.

7.4.9. Feladat. Írjon az LR transzformációt megvalósító eljárást.
Az eljárás paraméterezése `LRtransz(A,kMAX,eps)` alakú legyen, ahol

`LRtransz`	az eljárás neve,
`A`	a vizsgálandó $\mathbf{A}$ mátrix,
`kMAX`	az iterációs lépések maximális száma,
`eps`	a hibabecslésre használt konstans:

$$\max_{j,k=1,\ j>k}^{n} |a_{jk}| < eps.$$

7.4.10. Feladat. Alkalmazza a 7.4.9. Feladat programját a fejezetben szereplő **A** mátrixok sajátértékeinek közelítésére.

7.4.11. Feladat. Írjon az eltolásos QR transzformációt megvalósító eljárást.
Az eljárás paraméterezése `QRtranszS(A,kMAX,eps)` alakú legyen, ahol

`QRtranszS`	az eljárás neve,
`A`	a vizsgálandó **A** mátrix,
`kMAX`	az iterációs lépések maximális száma,
`eps`	a hibabecslésre használt konstans:

$$\max_{j,k=1,\ j>k}^{n} \left|a_{jk}\right| < eps.$$

7.4.12. Feladat. Alkalmazza a 7.4.11. Feladat programját a fejezetben szereplő mátrixok sajátértékeinek közelítésére.

7.4.13. Feladat. Közelítse az alábbi szimmetrikus **A** mátrixok sajátértékeit a Jacobi-forgatás különböző (maximális elemet nullázó, ciklikus, küszöbszámos) változataival. Hasonlítsa össze a kapott eredmények pontosságát. (Célszerű lebegőpontos értékekkel számolni.)

$$\text{a)}\begin{pmatrix} -3 & 3 & -3 & 0 \\ 3 & -4 & -2 & 4 \\ -3 & -2 & -3 & -3 \\ -3 & -2 & -3 & -3 \\ 0 & 4 & -3 & -2 \end{pmatrix}, \quad \text{b)}\begin{pmatrix} 0 & 2 & 0 & 2 \\ 2 & -1 & 2 & -1 \\ 0 & 2 & -2 & -2 \\ 2 & -1 & -2 & 0 \end{pmatrix}, \quad \text{c)}\begin{pmatrix} 1 & -1 & -1 & 1 \\ -1 & 0 & -1 & 1 \\ -1 & -1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}.$$

7.4.14. Feladat. Ha a sajátértékek (numerikus) meghatározásakor sikerül megkapnunk az $\mathbf{A} \in \mathbb{R}^{n\times n}$ mátrix λ_1 sajátértékét és egy hozzá tartozó $\mathbf{x}_1$ sajátvektort, akkor természetesnek tűnik, hogy csökkentsük a feladat dimenzióját, s a további sajátérték/sajátvektor párokat eggyel kisebb dimenziós mátrixokkal dolgozva határozzuk meg. Ilyen mátrixok a *rangszámcsökkentéseknek* nevezett eljárásokkal állíthatók elő. Vizsgálja a következő ortogonális rangszámcsökkentő módszert.
Föltéve, hogy ismert az $\mathbf{A} \in \mathbb{R}^{n\times n}$ mátrix $\lambda_1 \in \mathbb{R}$, $\mathbf{x}_1 \in \mathbb{R}^n$ valós sajátérték/sajátvektor párja, legyen **H** az az elemi tükröző mátrix, melyre $\mathbf{H}\mathbf{x}_1 = \lambda_1\mathbf{x}_1$. Ortogonális hasonlósági transzformációval képezze az $\hat{\mathbf{A}} = \mathbf{H}\mathbf{A}\mathbf{H}^{\mathrm{T}}$ mátrixot.
Mutassa meg, hogy

$$\hat{\mathbf{A}} = \begin{pmatrix} \lambda_1 & \mathbf{b}_1^{\mathrm{T}} \\ \mathbf{0} & \mathbf{B} \end{pmatrix}$$

alakú lesz; **B** sajátértékei $\lambda_2, \lambda_3 \ldots, \lambda_n$, és a **B** mátrix λ_2-höz tartozó $\mathbf{y}_2$ sajátvektorának ismeretében könnyen megadható az **A** megfelelő $\mathbf{x}_2$ sajátvektora.

7.4.15. Feladat. Ha az LR transzformációt olyan speciális $\mathbf{T}_1$ tridiagonális mátrixszal kezdjük el, melynek a főátló feletti mellékátlójában mindenütt 1 áll, akkor a $\mathbf{T}_1 = \mathbf{L}_1\mathbf{R}_1$ felbontás alapján a következő azonosság írható fel

$$\mathbf{T}_1 = \begin{pmatrix} r_1^{(1)} & 1 & & & \\ l_1^{(1)} r_1^{(1)} & l_1^{(1)} + r_2^{(1)} & 1 & & \\ & l_2^{(1)} r_2^{(1)} & l_2^{(1)} + r_3^{(1)} & \ddots & \\ & & \ddots & \ddots & 1 \\ & & & l_{n-1}^{(1)} r_{n-1}^{(1)} & l_{n-1}^{(1)} + r_n^{(1)} \end{pmatrix} =$$

$$= \mathbf{L}_1 \mathbf{R}_1 = \begin{pmatrix} 1 & & & & \\ l_1^{(1)} & 1 & & & \\ & l_2^{(1)} & \ddots & & \\ & & \ddots & \ddots & \\ & & & l_{n-1}^{(1)} & 1 \end{pmatrix} \begin{pmatrix} r_1^{(1)} & 1 & & & \\ & r_2^{(1)} & 1 & & \\ & & r_3^{(1)} & \ddots & \\ & & & \ddots & 1 \\ & & & & r_n^{(1)} \end{pmatrix}.$$

Ekkor a $\mathbf{T}_2 = \mathbf{R}_1 \mathbf{L}_1$ mátrix elemei:

$$\mathbf{T}_2 = \begin{pmatrix} l_1^{(1)} + r_1^{(1)} & 1 & & & \\ l_1^{(1)} r_2^{(1)} & l_2^{(1)} + r_2^{(1)} & 1 & & \\ & l_2^{(1)} r_3^{(1)} & l_3^{(1)} + r_3^{(1)} & \ddots & \\ & & \ddots & \ddots & 1 \\ & & & l_{n-1}^{(1)} r_n^{(1)} & r_n^{(1)} \end{pmatrix}$$

Mutassa meg, hogy ha adottak a $\mathbf{T}_j$ tridiagonális mátrix fönti indexezésnek megfelelő elemei, akkor az

$$r_k^{(j)} + l_k^{(j)} = r_k^{(j+1)} + l_{k-1}^{(j+1)}$$
$$r_{k+1}^{(j)} l_k^{(j)} = r_k^{(j+1)} + l_k^{(j+1)}$$

egyenlőségekből egyszerűen – az ún. QD algoritmus vagy rombusz-szabály segítségével – kifejezhetők a $\mathbf{T}_{j+1}$ (szintén tridiagonális) mátrix elemei.

7.4.16. Feladat. Alkalmazza a 7.4.15. Feladat algoritmusát az alábbi mátrixok sajátértékeinek közelítésére:

$$\begin{pmatrix} 2 & 1 & 0 & 0 & 0 \\ 2 & 4 & 1 & 0 & 0 \\ 0 & 4 & 3 & 1 & 0 \\ 0 & 0 & 4 & 2 & 1 \\ 0 & 0 & 0 & 3 & 4 \end{pmatrix}, \quad \begin{pmatrix} 2 & 1 & 0 & 0 & 0 \\ 1 & 3 & 1 & 0 & 0 \\ 0 & 4 & 4 & 1 & 0 \\ 0 & 0 & 2 & 2 & 1 \\ 0 & 0 & 0 & 3 & 3 \end{pmatrix}, \quad \begin{pmatrix} 5 & 1 & 0 & 0 & 0 \\ 3 & 2 & 1 & 0 & 0 \\ 0 & 4 & 3 & 1 & 0 \\ 0 & 0 & 2 & 2 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix}.$$

7.5. Sajátértékek perturbációja

7.5.1. Feladat. Ha $\hat{\mathbf{x}}$ és $\hat{\lambda}$ az $\mathbf{A}$ *közelítő* sajátérték-sajátvektor párja, tehát $\mathbf{A}\hat{\mathbf{x}} - \hat{\lambda}\hat{\mathbf{x}} = \mathbf{r} \neq \mathbf{0}$ és $\|\hat{\mathbf{x}}\|_2 = 1$, akkor milyen egyszerű (1 rangú) $\hat{\Delta}$ mátrixra lehet az $\mathbf{A} + \hat{\Delta}$ perturbált mátrixnak *pontos* sajátérték-sajátvektor párja $\hat{\mathbf{x}}$ és $\hat{\lambda}$?

7.5.2. Feladat. Hasonlítsa össze az alábbi $\mathbf{A}(\varepsilon) \in \mathbb{R}^{2\times 2}$, és az $\varepsilon = 0$ esetben adódó $\mathbf{A}(0) = \mathbf{I}$ egységmátrix sajátértékeit és sajátvektorait. Mit tapasztal?

$$\mathbf{A}(\varepsilon) = \begin{pmatrix} 1 & \varepsilon \\ 0 & 1 \end{pmatrix}.$$

7.5.3. Feladat. Határozza meg az alábbi $\mathbf{B}(\varepsilon) \in \mathbb{R}^{2\times 2}$ és az $\varepsilon = 0$ esetben adódó $\mathbf{B}(0)$ mátrix sajátértékeit és sajátvektorait. Mit tapasztal?

$$\mathbf{B}(\varepsilon) = \begin{pmatrix} 0 & \varepsilon \\ 1 & 0 \end{pmatrix}.$$

7.5.4. Feladat. Vizsgálja meg az alábbi $\mathbf{A}(\varepsilon) \in \mathbb{R}^{n\times n}$ és az $\varepsilon = 0$ esetben adódó $\mathbf{A}(0)$ mátrix sajátértékeinek kapcsolatát! (Igazolja, hogy $p_{\mathbf{A}(\varepsilon)}(\lambda) = \det(\mathbf{A}(\varepsilon) - \lambda\mathbf{I}) = (1-\lambda)^n + \varepsilon$.
Mit kap ebből például az $n = 100$, $\varepsilon = 10^{-100}$ esetben?)

$$\mathbf{A}(\varepsilon) = \begin{pmatrix} 1 & 1 & 0 & \cdots & \cdots & 0 \\ 0 & 1 & \ddots & \ddots & & \vdots \\ \vdots & 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \vdots & & \ddots & 1 & 1 \\ \varepsilon & 0 & \cdots & \cdots & 0 & 1 \end{pmatrix}.$$

7.5.5. Feladat. Vizsgálja a következő $\mathbf{A}(\varepsilon) \in \mathbb{R}^{n\times n}$ mátrix sajátértékeit és karakterisztikus polinomját „kis pozitív" ε értékekre, illetve az $\varepsilon = 0$ esetben.
Igazolja, hogy $p_{\mathbf{A}(\varepsilon)}(\lambda) = \det(\mathbf{A}(\varepsilon) - \lambda\mathbf{I}) = (20-\lambda)(19-\lambda)\cdots(1-\lambda) - 20^{19}\varepsilon$.

$$\mathbf{A}(\varepsilon) = \begin{pmatrix} 20 & 20 & 0 & \cdots & \cdots & 0 \\ 0 & 19 & \ddots & \ddots & & \vdots \\ \vdots & 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \vdots & & \ddots & 2 & 20 \\ \varepsilon & 0 & \cdots & \cdots & 0 & 1 \end{pmatrix}.$$

7.5.6. Feladat. Vizsgálja a következő $\mathbf{A}_n \in \mathbb{R}^{n\times n}$ mátrix sajátértékeit és karakterisztikus polinomját. Mutassa meg, hogy a nagy sajátértékek jól, a kicsik gyengén meghatározottak.

$$\mathbf{A}_n = \begin{pmatrix} n & n-1 & n-2 & \cdots & 2 & 1 \\ n-1 & n-1 & n-2 & \cdots & 2 & 1 \\ 0 & n-2 & n-2 & \cdots & 2 & 1 \\ \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\ \vdots & & \ddots & 2 & 2 & 1 \\ 0 & \cdots & \cdots & 0 & 1 & 1 \end{pmatrix}.$$

7.5.7. Feladat. Vizsgálja a következő $\mathbf{A}_n \in \mathbb{R}^{2\times 2}$ mátrix sajátértékeit és sajátvektorait az $\alpha_n \to 0$ esetben.

$$\mathbf{A}_n = \begin{pmatrix} 1+\alpha_n \cos\dfrac{2}{\alpha_n} & -\alpha_n \sin\dfrac{2}{\alpha_n} \\ -\alpha_n \sin\dfrac{2}{\alpha_n} & 1-\alpha_n \cos\dfrac{2}{\alpha_n} \end{pmatrix}.$$

Milyen kapcsolatban vannak ezek a határérték-mátrix sajátértékeivel és sajátvektoraival?

8. fejezet

Közelítések lineáris terekben

8.1. Interpolációs közelítések

8.1.1. Feladat. Mutassa meg, hogy a $\mathcal{G}$ véges dimenziós altérre vonatkozó általánosított interpolációs feladatnak akkor és csak akkor létezik egyértelmű megoldása tetszőleges $x_j \in [a, b]$ alappontok és tetszőleges, az $[a, b]$ fölött értelmezett $f(x)$ függvény esetén, ha $\mathcal{G}$ Haar-altér, azaz tetszőleges $g(x) \in \mathcal{G}$ általánosított polinomnak legfeljebb $n-1$ zérushelye van $[a, b]$-ben.

8.1.2. Feladat. Mutassa meg, hogy tetszőleges, páronként különböző $\alpha_1, \ldots, \alpha_n$ valós számok esetén $\mathcal{G} = \mathrm{span}\,(e^{\alpha_1 x}, e^{\alpha_2 x}, \ldots, e^{\alpha_n x})$ Haar-altér a teljes $(-\infty, +\infty)$ számegyenesen.

8.1.3. Feladat. Mutassa meg, hogy $\mathcal{G} = \mathrm{span}\left(\frac{1}{p(x)}, \frac{x}{p(x)}, \ldots, \frac{x^{n-1}}{p(x)}\right)$ Haar-altér minden olyan valós intervallumon, amelyben a $p(x) \in \mathbb{R}[x]$ valós polinomnak nincsen gyöke.

8.1.4. Feladat. Mutassa meg, hogy ha a $g(x)$ függvény szigorúan monoton $[a, b]$-n, akkor $\mathcal{G} = \mathrm{span}\left(1, g(x), (g(x))^2, \ldots, (g(x))^{n-1}\right)$ Haar-altér az $[a, b]$ intervallumon.

8.1.5. Feladat. Mutassa meg, hogy tetszőleges $[a, b]$ intervallum és $\rho(x)$ súlyfüggvény esetén a hozzájuk tartozó ortogonális polinomrendszer első n elemére
$\mathcal{G} = \mathrm{span}\,(p_0(x), p_1(x), p_2(x), \ldots, p_{n-1}(x))$ Haar-altér az $[a, b]$ intervallumon.

8.1.6. Feladat. Határozza meg az $f(x) = x \sin x$ függvénynek a $[-1, 1]$ intervallumon fölvett $x_k = -1 + k\frac{2}{n}$, $k = 0, 1, \ldots, n$ alappontokhoz tartozó általánosított interpolációs polinomját a $\mathcal{G} = \mathrm{span}\,(P_0(x), P_1(x), \ldots, P_n(x))$ Legendre-polinomok által generált altérben.

8.1.7. Feladat. Határozza meg az $f(x) = xe^x$ függvénynek az $(\frac{1}{5}, \frac{2}{5}, \frac{3}{5}, \frac{4}{5}, 1)$ alappontokhoz tartozó általánosított interpolációs polinomját a $[0, 1]$ intervallumon a
$\mathcal{G} = \mathrm{span}\left(1, e^x, e^{2x}, e^{3x},\ e^{4x}\right)$ függvények által generált altérben.

8.1.8. Feladat. Írjon az általánosított interpolációt megvalósító (Maple) eljárást.
Az eljárás paraméterezése `AltInt(G, f, X)` alakú legyen, ahol

`AltInt`	az eljárás neve,
`G`	az alteret generáló függvények vektora,
`f`	a közelítendő $f(x)$ függvény,
`X`	az alappontok vektora.

8.1.9. Feladat. Használja a 8.1.8. Feladat programját a fejezet korábbi feladatainak megoldására.

8.1.10. Feladat. Mutassa meg, hogy az

x_i	f_i
0	1/3
1	1
2	1

adatokkal definiált racionális interpolációs feladatnak nincs $r_{1,1}(x) = \dfrac{\alpha_0 x + \alpha_1}{\beta_0 x + \beta_1}$ alakú megoldása. A számláló és a nevező fokszámát megadó számpárt, jelen esetben az $(1,1)$-et, a továbbiakban az interpolációs feladat *típusának* hívjuk.

8.1.11. Feladat. Mutassa meg, hogy az előző adatokat egy újabb alapponttal kiegészítve:

x_i	f_i
0	1/3
1	1
2	1
3	2/3

az $(n, m) = (1,2)$ típusú racionális interpolációs feladat már egyértelműen megoldható.

8.1.12. Feladat. Írja fel inverz differenciákkal az

x_i	f_i
0	3
1	1
2	1
3	3/2

értékek által meghatározott $r_{2,1}(x)$ racionális interpolációs polinomot.

8.1.13. Feladat. Mutassa meg, hogy az

x_i	f_i
-1	-1
0	2
1	1
2	$\frac{1}{2}$

$(n, m) = (1,2)$ típusú racionális interpolációs feladat nem oldható meg.

8.1.14. Feladat. Oldja meg az előző feladatokat a Maple `CurveFitting` csomagjának `RationalInterpolation`, illetve `ThieleInterpolation` eljárásaival.

8.1.15. Feladat. A komplex függvénytan egyik alaptétele szerint a ∞ értékkel kiegészített $\mathbb{C}^\infty$ komplex számtesten értelmezett bármely $r(x) = \frac{\alpha_0 x+\alpha_1}{\beta_0 x+\beta_1}$ törtlineáris függvény vagy kölcsönösen egyértelmű, vagy konstans.
Mi következik ebből az $(1,1)$ típusú racionális interpolációs feladatok megoldhatóságára vonatkozóan?

8.1.16. Feladat. Közelítse az $f(x)=\mathrm{tg}(x)$ függvényt a $[-\pi,\pi]$ intervallumon ekvidisztánsan felvett $(-\pi, \frac{-2}{3}\pi, \frac{-1}{3}\pi, 0, \frac{1}{3}\pi, \frac{2}{3}\pi, \pi)$ alappontokhoz tartozó $r_{3,3}(x) = \frac{p_3(x)}{q_3(x)}$ alakú racionális interpolációs polinommal.

8.1.17. Feladat. Közelítse az $f(x) = x+\sin(x^2)$ függvényt a $[0,2\pi]$ intervallumon ekvidisztánsan felvett $(0, \frac{2}{7}\pi, \frac{4}{7}\pi, \frac{6}{7}\pi, \frac{8}{7}\pi, \frac{10}{7}\pi, \frac{12}{7}\pi)$ alappontokhoz tartozó $t(x)$ trigonometrikus interpolációs polinommal.

8.2. Legjobb közelítések lineáris terekben

8.2.1. Feladat. Igazolja, hogy tetszőleges normált lineáris $\mathcal{V}$ térben a háromszög egyenlőtlenség két szokásos formája, miszerint bármely $\mathbf{f}, \mathbf{h} \in \mathcal{V}$ esetében

$$\|\mathbf{f}+\mathbf{h}\| \le \|\mathbf{f}\|+\|\mathbf{h}\| \quad \text{illetve} \quad \|\mathbf{f}-\mathbf{h}\| \ge |\|\mathbf{f}\|-\|\mathbf{h}\||$$

ekvivalens egymással.

8.2.2. Feladat. A $\mathcal{V}$ lineáris tér *szigorúan normált*, ha tetszőleges, $\mathbf{0}$-tól különböző $\mathbf{f}, \mathbf{h} \in \mathcal{V}$ elemekre a háromszög-egyenlőtlenségben fönnálló

$$\|\mathbf{f}+\mathbf{h}\| = \|\mathbf{f}\|+\|\mathbf{h}\|$$

egyenlőségből következik, hogy a két elem lineárisan függő, vagyis $\alpha\mathbf{f}+\mathbf{h} = \mathbf{0}$ valamely $0 \neq \alpha \in \mathbb{R}$-rel.
Igazolja, hogy a folytonos függvények $\mathsf{C}[-1,1]$ tere az $\|f(x)\| = \max\limits_{0\le x\le 1} |f(x)|$ maximum- vagy Csebisev-normával nem szigorúan normált.

8.2.3. Feladat. A $\mathcal{V}$ téren értelmezett $\|.\|$ norma *szigorúan konvex*, ha a tér tetszőleges $\mathbf{f} \neq \mathbf{h}$ elemeire teljesül

$$\|\mathbf{f}\| = \|\mathbf{h}\| = 1 \quad \Longrightarrow \quad \left\|\frac{\mathbf{f}+\mathbf{h}}{2}\right\| < 1.$$

Mutassa meg, hogy ha a $\mathcal{V}$ téren értelmezett $\|.\|$ norma szigorúan konvex, akkor $\mathcal{V}$ tetszőleges $\mathbf{f}$ elemének *legfeljebb egy* legjobb közelítése létezhet a $\mathcal{V}$ valamely véges dimenziós $\mathcal{G}$ alterében.

8.2.4. Feladat. Megfordítható-e a 8.2.3. Feladat állítása?

8.2.5. Feladat. Mutassa meg, hogy a szigorúan konvex normának a 8.2.3. Feladatban megadott definíciója ekvivalens a következővel:
A $\mathcal{V}$ téren értelmezett $\|.\|$ norma *szigorúan konvex*, ha a tér tetszőleges $\mathbf{f} \neq \mathbf{h}$ elemeire és tetszőleges $\alpha \in [0,1]$-re teljesül

$$\|\mathbf{f}\| = \|\mathbf{h}\| = 1 \quad \Longrightarrow \quad \|\alpha\mathbf{f}+(1-\alpha)\mathbf{h}\| < 1.$$

8.2.6. Feladat. Mutassa meg, hogy tetszőleges $\mathcal{V}$ normált lineáris tér akkor és csak akkor szigorúan normált, ha a $\|.\|$ norma szigorúan konvex norma.

8.3. Négyzetesen legjobb közelítések lineáris terekben

8.3.1. Feladat. Mutassa meg, hogy a $\mathsf{C}[a, b]$ tér az $\langle f(x), g(x)\rangle = \int_a^b f(x)g(x)\,dx$ belső szorzatra nézve nem teljes.

8.3.2. Feladat. Bizonyítsa be a paralelogramma oldalainak és átlóinak négyzetösszegéről szóló ismert összefüggés következő általánosítását.

Állítás. *A $\mathcal{V}$ normált vektortér akkor és csak akkor lineáris belsőszorzat-tér, ha tetszőleges* $\mathbf{f}$ *és* $\mathbf{h}$ *elemeire az*

$$\|\mathbf{f}+\mathbf{h}\|^2 + \|\mathbf{f}-\mathbf{h}\|^2 = 2(\|\mathbf{f}\|^2 + \|\mathbf{h}\|^2)$$

egyenlőség teljesül.

8.3.3. Feladat. Igazolja, hogy a $\mathcal{V} = \mathbb{R}^n$ lineáris tér az $\|\mathbf{x}\|_p = \sqrt[p]{\sum_{j=1}^n |x_j|^p}$, $p > 1$ normával akkor és csak akkor lineáris belsőszorzat-tér, ha $p = 2$.

8.3.4. Feladat. Bizonyítsa be a lineáris belsőszorzat-terekben érvényes Beppo Levi egyenlőtlenséget, mely szerint tetszőleges $\mathcal{G} \subset \mathcal{V}$ altérre, tetszőleges $\mathbf{g}_1, \mathbf{g}_2 \in \mathcal{G}$ és tetszőleges $\mathbf{x} \in \mathcal{V}$, $\mathbf{x} \notin \mathcal{G}$ elemekre a $d = \inf_{\mathbf{g}\in\mathcal{G}} \|\mathbf{x}-\mathbf{g}\|$ jelöléssel

$$\|\mathbf{g}_1 - \mathbf{g}_2\| \le \sqrt{\|\mathbf{x}-\mathbf{g}_1\|^2 - d^2} + \sqrt{\|\mathbf{x}-\mathbf{g}_2\|^2 - d^2}.$$

Az egyenlőtlenségnek azt a speciális esetét igazolja geometriai eszközökkel, amikor $\mathcal{V} = \mathbb{R}^3$, $\mathcal{G} = \mathrm{span}(\mathbf{g}_1, \mathbf{g}_2)$ kétdimenziós altér, a norma pedig a szokásos $\|.\|_2$ euklideszi norma.

8.3.5. Feladat. Igazolja, hogy ha $\mathbf{g}_1, \mathbf{g}_2, \ldots, \mathbf{g}_n$ a $\mathcal{V}$ lineáris belsőszorzat-tér $\mathcal{G}$ alterének tetszőleges bázisa, akkor akkor a megfelelő $\mathbf{G}$ Gram mátrix pozitív definit.

8.3.6. Feladat. Határozza meg az $[a, b]$ intervallumon folytonos $f(x)$ függvényt négyzetesen legjobban közelítő 0-ad fokú (konstans) polinomot a $\rho(x) \equiv 1$ súlyfüggvényre vonatkozóan.

8.3.7. Feladat. Határozza meg az $f(x) = e^x$ függvényt négyzetesen legjobban közelítő legfeljebb másodfokú polinomot a $[-1, +1]$ intervallumon. (Tehát $[a, b] = [-1, +1]$, $\rho(x) \equiv 1$ és $\mathcal{G} = \mathrm{span}(1, x, x^2)$.

8.4. Ortogonális polinomrendszerek

8.4.1. Feladat. Tekintse a $[-1,1]$ intervallumon azt az összes $p(x)$ polinom által alkotott lineáris belsőszorzat-teret, amelyet a $\rho(x) = x^2$ súlyfüggvénnyel definiált

$$\langle p, g\rangle = \int_{-1}^{1} p(x)q(x)x^2\,dx$$

belső szorzat határoz meg. Milyen β értékre lesznek a $p(x) = \beta x^2 + 1$ és a $q(x) = x + 1$ polinomok ortogonálisak?

8.4.2. Feladat. Határozza meg adott $[a,b]$ intervallum és $\rho(x)$ súlyfüggvény esetén az n-dik ortogonális polinom felírásához szükséges belsőszorzat-kiértékelések számát

a) a Gram-Schmidt féle ortogonalizációs képletek alapján: $p_0(x) \equiv 1$, tetszőleges $n > 0$-ra pedig

$$p_n(x) = x^n + \alpha_0 p_0(x) + \alpha_1 p_1(x) + \cdots \alpha_{n-1} p_{n-1}(x)$$

alakú, ahol

$$\alpha_i = -\frac{\langle x^i, p_i\rangle}{\langle p_i, p_i\rangle} \quad i = 0,1,\ldots,n-1.$$

b) a Láncos-féle rekurzió használatával: $p_0(x) \equiv 1$, $p_1(x) = x - \beta_1$, tetszőleges $n > 1$-re pedig

$$p_{n+1}(x) = (x-\beta_n)p_n(x) - \gamma_n^2 p_{n-1}(x),$$

ahol a konstansokat a

$$\beta_n = \frac{\langle xp_n, p_n\rangle}{\langle p_n, p_n\rangle} \quad \text{ha} \quad n \geq 0$$

és a

$$\gamma_1^2 = 0, \quad \gamma_n^2 = \frac{\langle p_n, p_n\rangle}{\langle p_{n-1}, p_{n-1}\rangle} \quad \text{ha} \quad n \geq 1$$

képletek definiálják.

8.4.3. Feladat. Ismert eredmény, hogy az ortogonális polinomrendszerek minden tagjának minden gyöke valós, egyszeres, továbbá a nyitott (a,b) intervallumba esik. Továbbá $p_{n+1}(x)$ gyökei szétválasztják $p_n(x)$ gyökeit bármely $n \geq 1$-re.
A föntieket az intervallumfelezéssel kombinálva készítsen olyan rekurzív eljárást, amellyel adott ε pontossággal meghatározható $p_n(x)$ összes gyöke.
Próbálja is ki a módszert az ismertebb ortogonális polinomrendszerek elemeire, például a Legendre- vagy a Csebisev-polinomokra.

8.4.4. Feladat. Milyen kapcsolat van a 8.4.2. Feladatban a Láncos-féle rekurzióval megadott $p_k(x)$ ortogonális polinomok és a

$$\mathbf{T}_k = \begin{pmatrix} \beta_1 & -\gamma_2 & 0 & \cdots & \cdots & 0 \\ -\gamma_2 & \beta_2 & -\gamma_3 & 0 & \cdots & 0 \\ 0 & -\gamma_3 & \beta_3 & -\gamma_4 & \ddots & \vdots \\ \vdots & 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & -\gamma_k \\ 0 & 0 & \cdots & 0 & -\gamma_k & \beta_k \end{pmatrix}$$

tridiagonális mátrixok között? Mire használható ez a kapcsolat?

8.4.5. Feladat. Bizonyítsa be, hogy ha a $\rho(x)$ súlyfüggvény $\frac{a+b}{2}$-re nézve páros függvény (például $[a,b] = [-1,1]$ és $\rho(-x) = \rho(x)$), akkor az összes $p_{2k}(x)$ ortogonális polinom is páros, az összes $p_{2k+1}(x)$ ortogonális polinom pedig páratlan függvény $\frac{a+b}{2}$-re nézve.

8.4.6. Feladat. Írjon olyan (Maple) eljárást, amely adott $[a, b]$ intervallum és $\rho(x)$ súlyfüggvény esetén a Gram-Schmidt féle ortogonalizációs képletek segítségével kiszámolja a megadott intervallumhoz és súlyfüggvényhez tartozó ortogonális polinomrendszer első n elemét. Az eljárás paraméterezése `OrtoPoliG(a,b,rho,n)` alakú legyen, ahol

`OrtoPoliG`	az eljárás neve,
`a`	az intervallum bal végpontja,
`b`	az intervallum jobb végpontja,
`rho`	a súlyfüggvényt megadó Maple kifejezés,
`n`	a meghatározandó polinomok száma.

8.4.7. Feladat. Írjon olyan (Maple) eljárást, amely adott $[a, b]$ intervallum és $\rho(x)$ súlyfüggvény esetén a Láncos-féle rekurziós képletek segítségével kiszámolja az adott intervallumhoz és súlyfüggvényhez tartozó ortogonális polinomrendszer első n elemét.
Az eljárás paraméterezése `OrtoPoliL(a,b,rho,n)` alakú legyen, ahol

`OrtoPoliL`	az eljárás neve,
`a`	az intervallum bal végpontja,
`b`	az intervallum jobb végpontja,
`rho`	a súlyfüggvényt megadó Maple kifejezés,
`n`	a meghatározandó polinomok száma.

8.4.8. Feladat. Bizonyítsa be, hogy a $T_n(x)$ elsőfajú Csebisev-polinomokra teljesülnek a következő azonosságok:

a) $T_{2n}(x) = 2\,(T_n(x))^2 - 1$,

b) $T_{m+n}(x) + T_{m-n}(x) = 2T_m(x)T_n(x)$,

c) $\dfrac{T_n'(x)}{n} = 2T_{n-1}(x) + \dfrac{T_{n-2}'(x)}{n-2}$,

d) $\dfrac{T_{2n}'(x)}{2n} = 2\,(T_{2n-1}(x) + T_{2n-3}(x) + \dots + 1)$.

8.4.9. Feladat. Adott $[a, b]$ intervallum és $\rho(x)$ súlyfüggvény esetén tetszőleges $f(x) \in$ $\in \mathsf{C}[a, b]$ függvényre tekintsük $f(x)$ azon $P_1(x)$, $P_2(x)$, ... Lagrange-polinomjainak sorozatát, melyben a $P_n(x)$ Lagrange-polinom alappontjai az adott intervallumhoz és súlyfüggvényhez tartozó ortogonális polinomrendszer n-dik elemének gyökei. Mutassa meg, hogy

$$\lim_{n\to\infty} \left\| f(x) - P_n(x) \right\|^2 = \lim_{n\to\infty} \int_a^b (f(x) - P_n(x))^2 \rho(x)\, dx = 0,$$

vagyis ezek az interpolációs polinomok az adott normában konvergálnak $f(x)$-hez.

8.4.10. Feladat. Bizonyítsa be a $p_n(x)$ n-dik ortogonális polinom $x_1, x_2, \dots, x_n$ zérushelyeihez tartozó $L_i(x)$ Lagrange-féle bázispolinomokra vonatkozó alábbi azonosságokat:

a) Ha $i \neq j$, akkor $\langle L_i, L_j \rangle = 0$, tehát az n darab $n-1$-ed fokú $L_i(x)$ bázispolinom páronként ortogonális;

b) $x_i = \frac{\langle xL_i, L_i\rangle}{\langle L_i, L_i\rangle}$ alakban is fölírható.

8.4.11. Feladat. Bizonyítsa be, hogy tetszőleges $[a, b]$ intervallum és $\rho(x)$ súlyfüggvény esetén az ezekhez tartozó ortogonális polinomrendszer polinomjainak gyökei mindenütt sűrűn helyezkednek el $[a, b]$-ben. A bizonyításhoz használja fel azt az ismert tételt (v.ö. 10.4.4. Feladat), hogy a Gauss kvadratúrák sorozata tetszőleges $f(x) \in \mathrm{C}[a, b]$ függvényre konvergens.

8.4.12. Feladat. Határozza meg az $f(x) = \sin x$, $\cos x$, e^x függvények $f(x) \approx \sum_{k=0}^{n} \alpha_k P_k(x)$ alakú, Legendre-polinomok szerinti sorfejtését a $[-1,1]$ intervallumon az $n = 1,2,3,4,5$ értékekre. Hasonlítsa össze a kapott közelítések pontosságát a megfelelő fokszámú, 0 körüli Taylor-polinomokéval.

8.4.13. Feladat. Határozza meg az $f(x) = \operatorname{sgn} x$ függvény $f(x) \approx \sum_{k=0}^{n} \alpha_k P_k(x)$ alakú, Legendre-polinomok szerinti sorfejtését a $[-1,1]$ intervallumon az $n = 1,2,3,4,5$ értékekre. Ábrázolja grafikusan a kapott közelítéseket. Mit tapasztal?

8.4.14. Feladat. Határozza meg az $f(x) = \operatorname{sgn} x$ függvény $f(x) \approx \sum_{k=0}^{n} \alpha_k T_k(x)$ alakú, Csebisev-polinomok szerinti sorfejtését a $[-1,1]$ intervallumon az $n = 1,2,3,4,5$ értékekre. Ábrázolja grafikusan a kapott közelítéseket. Hasonlítsa össze a kapott eredményeket a 8.4.13. Feladatéval. Mit tapasztal?

8.4.15. Feladat. Közelítse az $f(x) = \sqrt{1 - x^2}$ függvényt a $[-1,1]$ intervallumban a Csebisev-polinomok szerinti sorfejtését felhasználva 0.001 pontossággal. Hány tagot kell ehhez a pontossághoz figyelembe venni a sorfejtésből?

8.4.16. Feladat. Határozza meg az $f(x) = \sin x$, $\cos x$, e^x függvények $f(x) \approx \sum_{k=0}^{n} \alpha_k T_k(x)$ alakú, Csebisev-polinomok szerinti sorfejtését a $[-1,1]$ intervallumon az $n = 1,2,3,4,5$ értékekre. Hasonlítsa össze a kapott közelítések pontosságát a megfelelő fokszámú, 0 körüli Taylor-polinomokéval.

8.4.17. Feladat. Vizsgálja a $t_k(x) = \sum_{k=0}^{n} \frac{f^{(k)}(0)}{k!} x^k$ Taylor-polinomokat az $f(x) = \sin x$, $\cos x$, e^x függvények esetében az $n = 1,2,3,4,5$ értékekre a $[-1,1]$ intervallumon. Milyen $\tilde{t}_k(x)$ polinomok adódnak ezekből, ha az $1, x, x^2, \ldots, x^n$ hatványokat a Legendre-polinomokkal fejezi ki, vagyis $\tilde{t}_k(x) = \sum_{k=0}^{n} \beta_k P_k(x)$ alakra hozza?
Milyen előnyei/hátrányai lehetnek ennek az átalakított polinomnak az eredetihez képest?

8.4.18. Feladat. Vizsgálja a $t_k(x) = \sum_{k=0}^{n} \frac{f^{(k)}(0)}{k!} x^k$ Taylor-polinomokat az $f(x) = \sin x$, $\cos x$, e^x függvények esetében az $n = 1,2,3,4,5$ értékekre a $[-1,1]$ intervallumon. Milyen $\tilde{t}_k(x)$ polinomokat adódnak ezekből, ha az $1, x, x^2, \ldots, x^n$ hatványokat a Csebisev-polinomokkal fejezi ki, tehát $\tilde{t}_k(x) = \sum_{k=0}^{n} \gamma_k T_k(x)$ alakra hozza?
Milyen előnyei/hátrányai lehetnek ennek az átalakított polinomnak az eredetihez képest?

8.4.19. Feladat. Hogyan határozható meg $O(n)$ aritmetikai művelettel az $f(x)$ függvény $f(x) \approx \phi(x) = \sum_{k=0}^{n} \alpha_k p_k(x)$ ortogonális polinomok szerinti szerinti sorfejtésének $\phi(x_0)$ helyettesítési értéke adott $x_0 \in \mathbb{R}$-re?

8.5. Egyenletes közelítések

8.5.1. Feladat. Mi lesz az $f(x) \in \mathrm{C}[a, b]$ folytonos függvényt az $[a, b]$ intervallumon egyenletesen legjobban közelítő konstans függvény?

8.5.2. Feladat. Határozza meg az $f(x) = x^3$ függvényt a $[0,1]$ intervallumon egyenletesen legjobban közelítő $p_n^*(x)$ n-ed fokú polinomot $n = 0,1,2$ esetén.
Csebisev tételéből induljon ki: keressen $n+2$ elemű optimális alternáló pontsorozatot.

8.5.3. Feladat. Határozza meg az $f(x) = \sin 4x$ függvényt a $[0,2\pi]$ intervallumon egyenletesen legjobban közelítő k-ad fokú polinomokat a $k = 1,2,3,4$ értékekre.

8.5.4. Feladat. Határozza meg az $f_k(x)=x^k \sin x^k$ függvényeket a $[0,1]$ intervallumon egyenletesen legjobban közelítő $p_{2,k}^*(x)$ másodfokú polinomokat a $k = 1, \ldots, 10$ értékekre. (Felhasználhatja például a Maple `numapprox` csomagját.) Mit tapasztal?

8.5.5. Feladat. Mennyiben változik a helyzet a 8.5.4. Feladathoz képest, ha az $f_k(x)=x^k \sin x^k$ függvényt a $[0,1]$ intervallumon az egyenletesen legjobban közelítő k-ad fokú polinomjával közelíti?

8.5.6. Feladat. Mutassa meg, hogy ha az $f(x)$ függvény az $[a, b]$ intervallumon kétszer folytonosan differenciálható és második deriváltja jeltartó $[a, b]$-n, akkor az $f(x)$-et egyenletesen legjobban közelítő $p_1^*(x)$ elsőfokú polinom megszerkeszthető úgy, hogy vesszük $f(x)$-nek az a és b pontokhoz tartozó $P_1(x)$ elsőfokú Lagrange-polinomját, majd $f(x)$-nek a $P_1(x)$ egyenesével párhuzamos (egyetlen, jól definiált) $T_1(x)$ érintőjét, és a két egyenes távolságát megfelezve felrajzoljuk a (mindkettőjükkel párhuzamos) $p_1^*(x)$ egyenest.
A szerkesztés végrehajtásához természetesen szükség van egy $f'(x)=m$ alakú, általában nemlineáris egyenlet megoldására - ez történhet valamely közelítő módszerrel, vagy segítségül hívhatja a Maple-t.

8.5.7. Feladat. Hajtsa végre a 8.5.6. Feladatban leírt szerkesztést megfelelő $[a, b]$ intervallumot választva az ismert elemi függvényekre: $\sin x$, $\ln x$, $\sqrt{x}$, stb.
Készítsen grafikonokat is, s ennek alapján próbálja megítélni a kapott közelítéseket.

8.5.8. Feladat. Határozza meg az $f(x) = \ln(x+1)$ függvényt a $[\frac{-1}{2}, \frac{1}{2}]$ intervallumon egyenletesen legjobban közelítő $p_1^*(x)$ elsőfokú polinomot.

8.5.9. Feladat. Bizonyítsa be a legjobb közelítések unicitására vonatkozó következő állítást.

Állítás. *H a $\mathcal{G} \subset \mathrm{C}[a, b]$ altér Haar-altér, akkor tetszőleges $f(x) \in \mathrm{C}[a, b]$ függvénynek legfeljebb egy $p^*(x)$ legjobb közelítése lehet $\mathcal{G}$-ben.*

8.5.10. Feladat. Bizonyítsa be a következő állítást.

Állítás. *Ha $f(x)$ páros (páratlan) függvény a $[-a, a]$ intervallumon, akkor az $f(x)$-et egyenletesen legjobban közelítő k-ad fokú $p_k(x)$ polinom szintén páros (páratlan) függvény ezen az intervallumon.*

8.5.11. Feladat. A 8.5.10. Feladatot felhasználva határozza meg az $f(x) = \cos x$ függvényt egyenletesen legjobban közelítő harmadfokú polinomot az $[a,b] = [-\frac{\pi}{2}, \frac{\pi}{2}]$ intervallumon.

8.5.12. Feladat. Határozza meg az $f(x) = e^{x^2}$ függvényt egyenletesen legjobban közelítő harmadfokú polinomot a $[-1,1]$ intervallumon. Használja föl, hogy $f(x)$ páros függvény!

8.5.13. Feladat. Bizonyítsa be a következő állítást.

Állítás. *Legyen $f(x)$ folytonos $[a,b]$-n. Ekkor tetszőleges $n \geq 1$-re az $[a,b]$-ben tetszőlegesen felvett $n+1$ darab alapponthoz tartozó $P_n(x)$ Lagrange-polinom hibájára*

$$\max_{x\in[a,b]} \left|f(x) - P_n(x)\right| \leq (1+L_n)E_n(f)$$

teljesül, ahol

$$L_n = \max_{x\in[a,b]} \sum_{i=0}^{n} \left|L_i(x)\right|$$

az ún. Lebesgue-konstanst, $E_n(f)=\min(\left\| f(x) - p(x)\right\|_\infty \mid \deg(p) \leq n)$ pedig $f(x)$ legföljebb n-ed fokú polinommal való legjobb közelítésének hibáját jelöli.

8.5.14. Feladat. Milyen korlát vezethető le a 8.5.13.Feladat állításából a $[-1, 1]$ intervallumon felvett 10 Csebisev alapponthoz tartozó Lagrange-polinom hibájára?

8.5.15. Feladat. Igazolja, hogy tetszőleges $[a,b]$ intervallum, $\rho(x)$ súlyfüggvény és $n \geq 0$ esetében az $f(x) \in \mathrm{C}[a,b]$ függvényt *négyzetesen* legjobban közelítő $q^*{}_n(x)$ n-ed fokú polinom $\delta_n^2 = \int_a^b \left|f(x) - q_n^*(x)\right|^2 \rho(x)\,dx$ négyzetes hibájára, illetve az *egyenletesen* legjobban közelítő $p^*{}_n(x)$ n-ed fokú polinom $E_n(f) = \max_{x\in[a,b]} \left|f(x) - p_n^*(x)\right|$ hibájára a $\delta_n^2 \leq$ $\leq (E_n(f))^2 \int_a^b \rho(x)\,dx$ egyenlőtlenség teljesül.

8.5.16. Feladat. Milyen korlátok adódnak a 8.5.15. Feladat egyenlőtlenségéből az ismert ortogonális polinomrendszerekre?

8.5.17. Feladat. Igazolja, hogy tetszőleges $[a,b]$ intervallum és $f(x) \in \mathrm{C}^n[a,b]$ esetén az $f(x)$-et egyenletesen legjobban közelítő $p_{n-1}^*(x)$ $n-1$-ed fokú polinom $E_{n-1}(f) = \max_{x\in[a,b]} \left|f(x) - p_{n-1}^*(x)\right|$ hibájára teljesül az

$$\frac{1}{2^{n-1}n!} \min_{x\in[a,b]} \left|f^{(n)}(x)\right| \leq E_{n-1}(f) \leq \frac{1}{2^{n-1}n!} \max_{x\in[a,b]} \left|f^{(n)}(x)\right|$$

egyenlőtlenség.

8.5.18. Feladat. Milyen korlátok adódnak a 8.5.17. Feladat egyenlőtlenségéből az $f(x) = e^x$ függvényre?

9. fejezet

Egyenletrendszerek megoldása iterációs módszerekkel

9.1. Relaxációs és egyéb módszerek lineáris egyenletrendszerekre

9.1.1. Feladat. Az $\mathbf{Ax} = \mathbf{b}$, $\mathbf{A} \in \mathbb{R}^{n\times n}$, $\mathbf{b} \in \mathbb{R}^n$ nem-szinguláris együttható-mátrixú lineáris egyenletrendszer ekvivalens átalakításokkal az $\mathbf{x} = (\mathbf{I}-\mathbf{A})\mathbf{x}+\mathbf{b}$ alakra hozható.
Mutassa meg a Banach-féle fixponttétel felhasználásával, hogy ha $\|\mathbf{I}-\mathbf{A}\|_\infty < 1$, akkor az $\mathbf{x}_{k+1} = (\mathbf{I}-\mathbf{A})\mathbf{x}_k + \mathbf{b}$ iterációs képlettel számolt $\mathbf{x}_k$ közelítések az eredeti egyenletrendszer (egyetlen, létező) $\mathbf{x}^*$ megoldásához konvergálnak tetszőleges $\mathbf{x}_0 \in \mathbb{R}^n$ kezdővektor esetén.

9.1.2. Feladat. Tekintse az $\mathbf{Ax}=\mathbf{b}$, $\mathbf{A}\in\mathbb{R}^{n\times n}$, $\mathbf{b}\in\mathbb{R}^n$ pozitív definit együttható-mátrixú lineáris egyenletrendszer $\mathbf{x}=\left(\mathbf{I}-\frac{2}{\alpha}\mathbf{A}\right)\mathbf{x}+\frac{2}{\alpha}\mathbf{b}$ alakú ekvivalens átalakítását, ahol az α konstansra $\rho(\mathbf{A}) < \alpha$ teljesül.
Igazolja, hogy az $\mathbf{x}_{k+1} = \left(\mathbf{I}-\frac{2}{\alpha}\mathbf{A}\right)\mathbf{x}_k + \frac{2}{\alpha}\mathbf{b}$ átalakított egyenletrendszerre alkalmazott iterációs módszer globálisan konvergens.

9.1.3. Feladat. Ha az $\mathbf{Ax}=\mathbf{b}$ lineáris egyenletrendszert eliminációs módszerekkel, például az LR trianguláris felbontással oldottuk meg, akkor $\mathbf{LR}\approx\mathbf{A}$, tehát $\hat{\mathbf{A}}^{-1}=\mathbf{R}^{-1}\mathbf{L}^{-1}$ föltételezhetően az $\mathbf{A}^{-1}$ inverz jó közelítése. Erre támaszkodva az elimináció eredményét $\mathbf{x}_0$ kezdővektornak tekintve a következő vektorok kiszámításával végzett *utóiteráció* alkalmazható a megoldás pontosságának javítására.

a) $\mathbf{r}_k$ meghatározása: $\mathbf{r}_k = \mathbf{b}-\mathbf{A}\mathbf{x}_k$,

b) $\mathbf{z}_k$ meghatározása: $\mathbf{L}\mathbf{z}_k = \mathbf{r}_k$,

c) $\mathbf{u}_k$ meghatározása: $\mathbf{R}\mathbf{u}_k = \mathbf{z}_k$,

d) $\mathbf{x}_{k+1}$ meghatározása: $\mathbf{x}_{k+1} = \mathbf{x}_k+\mathbf{u}_k$.

Mennyi a fönti algoritmus műveletigénye, és milyen elegendő föltétel adható az $\mathbf{x}_k$ vektorok konvergenciájára?

9.1.4. Feladat. Az előző feladatok általánosításaként vizsgálja az $\mathbf{Ax} = \mathbf{b}$ egyenletrendszerre alkalmazott

$$\mathbf{x}_{k+1} = (\mathbf{I} - \mathbf{CA})\mathbf{x}_k + \mathbf{Cb}$$

alakú iterációt valamely rögzített $\mathbf{C} \in \mathbb{R}^{n\times n}$ mátrixszal. Milyen speciális eseteiről volt szó az előző feladatokban?

9.1.5. Feladat. Ha az $\mathbf{x} = \mathbf{Bx} + \mathbf{c}$ egyenletrendszerre alkalmazott $\mathbf{x}_{k+1} = \mathbf{Bx}_k + \mathbf{c}$ iteráció lassan, vagy egyáltalán nem konvergál, a helyzet néha javítható a kapott $\mathbf{x}_k$ közelítésekből számított „átlagolt, javított” értékekkel. Válasszon alkalmas $\tau \in \mathbb{R}^+$ *extrapolációs paramétert* és tekintse az

$$\mathbf{x}_{k+1} = \tau(\mathbf{Bx}_k + \mathbf{c}) + (1 - \tau)\mathbf{x}_k$$

képlettel definiált *extrapolált iterációs módszert.* Igazolja, hogy ha a $\mathbf{B}$ mátrix sajátértékeire teljesül

$$-\infty < \lambda_1 \le \lambda_2 \le \cdots \le \lambda_n < 1,$$

akkor a $\tau^* = \frac{2}{2-(\lambda_n+\lambda_1)}$ értékre az extrapolált módszer konvergens lesz, sőt ez lesz az optimális τ érték.

9.1.6. Feladat. Mit eredményez az extrapoláció a Jacobi iteráció esetében?

9.1.7. Feladat. Az $\mathbf{x} = \mathbf{Bx} + \mathbf{c}$ alakból kiindulva levezethető a $\mathbf{B}$ mátrix $\mathbf{B} = \mathbf{B}_1 + \mathbf{B}_2$ fölbontásán alapuló $\mathbf{x}_{k+1} = \mathbf{B}_1\mathbf{x}_{k+1} + \mathbf{B}_2\mathbf{x}_k + \mathbf{c}$ iterációs képlet.
Milyen elégséges feltétel adható ennek globális konvergenciájára?

9.1.8. Feladat. Hogyan alkalmazható a 9.1.7. Feladat a Gauss-Seidel iterációra?

9.1.9. Feladat. Mutassa meg, hogy az

$$\mathbf{A} = \begin{pmatrix} 10 & 1 & 1 & 1 \\ 10 & 10 & 1 & 1 \\ 10 & 10 & 10 & 1 \\ 10 & 10 & 10 & 10 \end{pmatrix}$$

együttható mátrix *nem* gyengén diagonális domináns, de a Gauss-Seidel iteráció konvergens.

9.1.10. Feladat. Igazolja, hogy ha az erősen reguláris együttható mátrixú $\mathbf{Ax} = \mathbf{b}$ lineáris egyenletrendszerre a Jacobi-iteráció konvergál, akkor alulrelaxálás esetén (azaz $0 < \omega < 1$-re) a Jacobi-féle relaxációs módszer, más néven a JOR-iteráció (lásd [13, 14]) is konvergálni fog.

9.1.11. Feladat. Ismert állítás, hogy ha az $\mathbf{Ax} = \mathbf{b}$ egyenletrendszer együtthatómátrixa gyengén diagonális domináns és irreducibilis, akkor mind a Jacobi-, mind a Gauss-Seidel iteráció konvergens. A következő példa azt mutatja, hogy nem lehet „kicsit” gyengíteni a feltételeken. Mutassa meg, hogy az

$$\mathbf{A} = \begin{pmatrix} 2 & 0 & -1 & -1 \\ 0 & 2 & -1 & -1 \\ -1 & -1 & 2 & 0 \\ -1 & -1 & 0 & 2 \end{pmatrix}$$

együtthatómátrixra már nem teljesül a konvergencia.

9.1.12. Feladat. Határozza meg, hogy milyen ω értékekre konvergál az $\mathbf{A}\mathbf{x}=\mathbf{b}$ egyenletrendszerre alkalmazott JOR-iteráció , ha

$$\mathbf{A}=\begin{pmatrix}4 & -1 & 0\\ -1 & 4 & -1\\ 0 & -1 & 4\end{pmatrix} \quad \text{és} \quad \mathbf{b}=\begin{pmatrix}1\\2\\1\end{pmatrix}.$$

9.1.13. Feladat. Vizsgálja a 4 iterációs módszer: Jacobi, Gauss-Seidel, JOR és SOR (más néven szukcesszív túlrelaxálás, lásd [13, 14]) konvergenciáját az $\mathbf{A}\mathbf{x}=\mathbf{b}$ egyenletrendszerre, ha $\mathbf{b}=(1,1,\ldots,1)^{\mathrm{T}}$ és

$$\mathbf{A}=\begin{pmatrix}3 & -1 & 0 & 0 & 0 & -1\\ -1 & 3 & -1 & 0 & -1 & 0\\ 0 & -1 & 3 & -1 & 0 & 0\\ 0 & 0 & -1 & 3 & -1 & 0\\ 0 & -1 & 0 & -1 & 3 & -1\\ -1 & 0 & 0 & 0 & -1 & 3\end{pmatrix},$$

a) elméleti eszközökkel (az ismert konvergenciatételek fölhasználásával),
b) gyakorlati számítási kísérletekkel.

9.1.14. Feladat. Az α paraméter mely értékeire lesz az $\mathbf{A}\mathbf{x}=\mathbf{b}$ egyenletrendszerre alkalmazott SOR-iteráció konvergens, ha az együttható mátrix

$$\mathbf{A}=\begin{pmatrix}1 & \alpha & \alpha\\ \alpha & 1 & \alpha\\ \alpha & \alpha & 1\end{pmatrix}$$

alakú?

9.1.15. Feladat. Vizsgálja a négy iterációs módszer konvergenciáját az α valós paraméter különböző értékeire az

$$\mathbf{A}=\begin{pmatrix}2 & 1 & \alpha\\ 1 & 2 & \alpha\\ \alpha & \alpha & 2\end{pmatrix}$$

együttható mátrixú lineáris egyenletrendszerekre.

9.1.16. Feladat. Írjon olyan (Maple) eljárást, amely adott $\mathbf{A}$ együttható mátrixú és $\mathbf{b}$ jobb oldalú lineáris egyenletrendszer megoldását a JOR-iterációval közelíti.
Az eljárás paraméterezése `LinJOR(A, b, x0, omega, kMAX, eps)` alakú legyen, ahol

`LinJOR`	az eljárás neve,
`A`	az $\mathbf{A}$ együttható mátrix
`b`	a $\mathbf{b}$ jobb oldali konstans vektor,
`x0`	az iteráció $\mathbf{x}_0$ kezdővektora,
`omega`	az ω relaxációs paraméter
`kMAX`	az iterációs lépések maximális száma,
`eps`	a k-dik közelítés maximális megengedett hibája: $\|\mathbf{b}-\mathbf{A}\mathbf{x}_k\| < eps$.

9.1.17. Feladat. Közelítse a 9.1.16. Feladatban megadott programmal a fejezet korábbi feladataiban szereplő lineáris egyenletrendszerek megoldásait.

9.1.18. Feladat. Írjon olyan (Maple) eljárást, amely adott **A** együttható mátrixú és **b** jobboldalú lineáris egyenletrendszer megoldását az SOR-iterációval közelíti.
Az eljárás paraméterezése `LinSOR(A, b, x0, omega, kMAX, eps)` alakú legyen, ahol

`LinSOR`	az eljárás neve,
`A`	az **A** együttható mátrix,
`b`	a **b** jobboldali konstans vektor,
`x0`	az iteráció $\mathbf{x}_0$ kezdővektora,
`omega`	az ω relaxációs paraméter
`kMAX`	az iterációs lépések maximális száma,
`eps`	a k-dik közelítés maximális megengedett hibája: $\\|\mathbf{b}-\mathbf{A}\mathbf{x}_k\\| < eps$.

9.1.19. Feladat. Közelítse a 9.1.18. Feladatban megadott programmal a fejezet korábbi feladataiban szereplő lineáris egyenletrendszerek megoldásait.

9.2. Fixpontiteráció

9.2.1. Feladat. Vizsgálja a következő egyenletrendszerek megoldhatóságát, a megoldások számát az α, β valós paraméterek különböző értékeire.

$$\text{a)}\quad \begin{aligned} \sin x_1 - x_2 &= 0\\ \alpha x_1 - x_2 + \beta &= 0, \end{aligned} \qquad \text{b)}\quad \begin{aligned} x_1^2 - x_2 + \alpha &= 0\\ -x_1 + x_2^2 - \alpha &= 0, \end{aligned} \qquad \text{c)}\quad \begin{aligned} \frac{1}{2}x_1 \sin\left(\frac{1}{2}\pi x_1\right) - x_2 &= 0\\ -x_1 + x_2^2 + 1 &= 0. \end{aligned}$$

9.2.2. Feladat. Mutassa meg, hogy a

$$\mathbf{G}(x_1, x_2) = \begin{pmatrix} \frac{1}{6}e^{-(x_1^2+x_2^2)} \\ \frac{1}{6}(x_1^2 - x_2^2) \end{pmatrix}$$

függvény kontrakció a $D = \left\{(x_1, x_2)\middle|0 \le x_1, x_2 \le 1\right\}$ halmazon. Határozza meg **G** fixpontját D-ben fixpontiterációval az $\left(\frac{1}{6}, \frac{1}{6^3}\right)^{\mathrm{T}}$ kezdőértékkel elindulva.

9.2.3. Feladat. Vizsgálja a kontrakciós tulajdonság teljesülését a

$$\mathbf{G}(x_1, x_2) = \frac{1}{2}\begin{pmatrix} 1 + \frac{\sin x_1}{4} + x_2 \\ 1 + \sin x_2 + x_1 \end{pmatrix}$$

függvény esetén a $\|.\|_\infty$ és a $\|.\|_2$ normákra vonatkozóan. Határozza meg **G** fixpontját az $\mathbf{x}_0 = (0, 0)^{\mathrm{T}}$ kezdőértékkel elindulva.

9.2.4. Feladat. Bizonyítsa be, hogy ha az

$$\begin{aligned} x_1 &= G_1(x_1, x_2),\\ x_2 &= G_2(x_1, x_2) \end{aligned}$$

egyenletrendszerre az $\mathbf{x}^*$ megoldás valamely D környezetében

$$\left|\frac{\partial G_1}{\partial x_1}\right| + \left|\frac{\partial G_2}{\partial x_1}\right| < 1 \quad \text{és} \quad \left|\frac{\partial G_1}{\partial x_2}\right| + \left|\frac{\partial G_2}{\partial x_2}\right| < 1$$

teljesül, akkor a fixpontiteráció tetszőleges $\mathbf{x}_0 \in D$ kezdőértéket véve konvergál az $\mathbf{x}^* \in D$ (D-ben egyetlen) megoldáshoz.

9.2.5. Feladat. Közelítse fixpontiterációval a következő egyenletrendszernek az egységnégyzetbe eső megoldását:

$$\begin{aligned} x_1 &= \frac{x_1^3 + x_2^3}{6} + \frac{1}{2}, \\ x_2 &= \frac{x_1^3 - x_2^3}{6} + \frac{1}{3}. \end{aligned}$$

9.2.6. Feladat. Közelítse fixpontiterációval a következő egyenletrendszer megoldását:

$$\begin{aligned} x_1 &= \frac{1}{2} \sin\left(\frac{1}{2}(x_1 - x_2)\right), \\ x_2 &= \frac{1}{2} \cos\left(\frac{1}{2}(x_1 + x_2)\right). \end{aligned}$$

Legyen $\mathbf{x}_0 = (-0.15, 0.5)^{\mathrm{T}}$.

9.2.7. Feladat. Közelítse fixpontiterációval a következő egyenletrendszer megoldását:

$$\begin{aligned} \sin x_1 - x_2 &= 1.5, \\ \cos x_2 - x_1 &= -0.5\,. \end{aligned}$$

Legyen $\mathbf{x}_0 = (1.4, -0.5)^{\mathrm{T}}$.

9.2.8. Feladat. Közelítse fixpontiterációval a következő egyenletrendszer megoldását:

$$\begin{aligned} \sin x_1 x_2 - \frac{x_2}{2\pi} - x_1 &= 0, \\ \left(1 - \frac{1}{4\pi}\right)\left(e^{2x_1} - e\right) + \frac{e x_2}{\pi} - 2e x_1 &= 0. \end{aligned}$$

Legyen $\mathbf{x}_0 = (0.4, 3)^{\mathrm{T}}$.

9.2.9. Feladat. Közelítse fixpontiterációval a következő egyenletrendszer megoldását. Grafikus ábrázolással keressen közelítő értékeket a megoldás(ok)ra.

$$\begin{aligned} x_1^2 + x_2^2 - 2x_1 &= 0, \\ x_1^2 - x_2 - \frac{1}{2} &= 0. \end{aligned}$$

9.2.10. Feladat. Közelítse fixpontiterációval a következő egyenletrendszer megoldását. A kezdővektor legyen $\mathbf{x}_0 = (2, 2, 2)^{\mathrm{T}}$.

$$
\begin{aligned}
x_1 &= \sin x_1 \sin x_2 \sin x_3 - \frac{1}{2}, \\
x_2 &= \frac{1}{4}\sqrt{x_1^2 + x_2^2 + x_3^2}, \\
x_3 &= \frac{1}{2}\left(x_1^2 + x_2^2\right).
\end{aligned}
$$

9.2.11. Feladat. Próbálja meg megoldani az előző egyenletrendszereket a Maple beépített `solve` vagy `fsolve` eljárásával is. Ahol lehetséges, először grafikus ábrázolással határozza meg a gyökök közelítő értékét.

9.2.12. Feladat. Írjon a fixpontiterációt megvalósító (Maple) eljárást.
Az eljárás paraméterezése `Fixpont(G,x0,kMAX,eps)` alakú legyen, ahol

`Fixpont`	az eljárás neve,
`G`	a megoldandó $\mathbf{x} = \mathbf{G}(\mathbf{x})$ egyenletrendszer jobb oldali függvényeiből álló vektor,
`x0`	az $\mathbf{x}_0$ kezdeti közelítés vektora,
`kMAX`	az iterációs lépések maximális száma,
`eps`	a k-dik közelítés maximális megengedett hibája.

9.2.13. Feladat. Próbálja meg megoldani az előző egyenletrendszereket a 9.2.12. Feladatban elkészített `Fixpont` eljárással. Hasonlítsa össze a kapott eredményeket a kézi számolás, vagy a Maple eredményeivel.

9.2.14. Feladat. Igazolja, hogy ha $\mathbf{G}(\mathbf{x})$ kontrakció az $\mathbf{x} = \mathbf{G}(\mathbf{x})$ egyenletrendszer $\mathbf{x}^*$ megoldásának valamely környezetében, akkor az $\mathbf{x}_{k+1} = \mathbf{G}(\mathbf{x}_k)$ fixpont-iteráció rendje legalább 1.

9.2.15. Feladat. Igazolja, hogy ha $1 \le p < r$ és az $\mathbf{x}_{k+1} = \mathbf{G}(\mathbf{x}_k)$ iteráció rendje legalább r, akkor az iteráció rendje legalább p. (Az iteráció rendje legalább p, ha létezik olyan α valós konstans, hogy $\|\mathbf{x}^* - \mathbf{x}_{k+1}\| \le \alpha \|\mathbf{x}^* - \mathbf{x}_k\|^p$) bármely k-ra.)

9.2.16. Feladat. Milyen egyszerű(bb), a gyakorlatban könnyebben alkalmazható elegendő feltétel vezethető le Ostrowski alábbi tételéből?

Állítás. *Ha az* $\mathbf{x} = \mathbf{G}(\mathbf{x})$ *egyenletrendszer* $\mathbf{x}^*$ *megoldásánál* $\mathbf{G}(\mathbf{x})$ *differenciálható és* $\mathbf{G}'(\mathbf{x}^*)$ *Jacobi mátrixára* $\rho(\mathbf{G}'(\mathbf{x}^*)) < 1$, *akkor az* $\mathbf{x}_{k+1} = \mathbf{G}(\mathbf{x}_k)$ *fixpont-iteráció lokálisan konvergál* $\mathbf{x}^*$*-hoz.*

9.2.17. Feladat. Alkalmazza a 9.2.16. Feladat eredményeit az előző egyenletrendszerek esetében a fixpont-iteráció lokális konvergenciájának vizsgálatára.

9.2.18. Feladat. Igazolja, hogy ha megadható olyan $\mathbf{x}^* \in \mathbb{R}^n$ vektor és $\{\alpha_k\}$ valós számsorozat, hogy $\lim_{k\to\infty} \alpha_k = \alpha$, $0 < \alpha < 1$, és a fixpontiterációval számított $\mathbf{x}_k$ vektorsorozatra $\|\mathbf{x}_{k+1} - \mathbf{x}^*\| \le \alpha_k \|\mathbf{x}_k - \mathbf{x}^*\|$, akkor $\lim_{k\to\infty} \mathbf{x}_k = \mathbf{x}^*$.

9.2.19. Feladat. Bizonyítsa be, hogy tetszőleges nem-szinguláris $\mathbf{A}$ mátrix esetében az

$$\mathbf{X}_{k+1} = \mathbf{X}_k + \mathbf{X}_k\,(\mathbf{I} - \mathbf{A}\mathbf{X}_k)$$

képlettel definiált $\mathbf{X}_k$ mátrixsorozat konvergál az $\mathbf{A}^{-1}$ inverz mátrixhoz, ha $\|\mathbf{I} - \mathbf{A}\mathbf{X}_0\| < 1$.

9.2.20. Feladat. Néha az $\mathbf{x}_{k+1} = \mathbf{G}(\mathbf{x}_k)$ fixpontiteráció konvergenciája az $\mathbf{x}_{k+1} = (1-\omega)\mathbf{x}_k + \omega\mathbf{G}(\mathbf{x}_k)$ nemlineáris relaxációval javítható.
Vizsgálja ennek lokális konvergenciáját a 9.2.16. feladatban említett Ostrowski-tétel segítségével.

9.2.21. Feladat. Az $\mathbf{F}(\mathbf{x}) = \mathbf{0}$ nemlineáris egyenletrendszer megoldását közelítse a reguláris szétvágások általánosításának tekinthető, Picard-iterációnak is nevezett módszerrel. Legyen $\mathbf{F}(\mathbf{x}) = \mathbf{0} \Longleftrightarrow \mathbf{R}\mathbf{x} - \mathbf{S}(\mathbf{x}) = \mathbf{0}$, ahol $\mathbf{R} \in \mathbb{R}^{n\times n}$ alkalmas nemszinguláris mátrix, s ennek alapján $\mathbf{x}_{k+1} = \mathbf{R}^{-1}\mathbf{S}(\mathbf{x}_k)$.
Vizsgálja a sorozat lokális konvergenciáját a 9.2.16. feladatban említett Ostrowski-tétel segítségével.

9.3. A Newton-módszer általánosításai

9.3.1. Feladat. Közelítse a következő egyenletrendszer megoldását a Newton-módszer valamelyik változatával:

$$\begin{aligned} 2x_1 + x_2 + + 4 &= 0 \\ 2x_1 + 3x_2 &= 0 \\ 3x_2 + x_2x_3 + 2 &= 0 \end{aligned}$$

9.3.2. Feladat. Határozza meg a többváltozós Newton-módszerrel a következő egyenletrendszer megoldását (megoldásait) az α paraméter 1 és 4 közötti értékei esetén:

$$\begin{aligned} \alpha x_1^3 - x_2^2 - 1 &= 0 \\ x_1 x_2^3 - x_2 - 4 &= 0 \end{aligned}$$

9.3.3. Feladat. Határozza meg a többváltozós Newton-módszerrel a következő egyenletrendszer megoldását (megoldásait):

$$\begin{aligned} \sin(x_1 - x_2) - x_1x_2 + 1 &= 0 \\ x_1^2 - x_2^2 - \frac{3}{4} &= 0 \end{aligned}$$

Próbálkozzon az egyszerűsített Newton-módszerrel is. Konvergál-e ennél a rendszernél?

9.3.4. Feladat. Határozza meg a Newton-módszer valamelyik változatával a következő egyenletrendszer megoldását (megoldásait):

$$\begin{aligned} \sin(x_1 + x_2) - \frac{3}{2}x_1 &= 0 \\ x_1^2 + x_2^2 - 1 &= 0 \end{aligned}$$

9.3.5. Feladat. Próbálja meg megoldani az előző egyenletrendszereket a Maple beépített `solve` vagy `fsolve` eljárásával. Ahol lehetséges, először grafikus ábrázolással határozza meg a gyökök közelítő értékét.

9.3.6. Feladat. Mutassa meg, hogy az $\mathbf{F}(\mathbf{x}) = \mathbf{A}\mathbf{x} - \mathbf{b} = 0$ lineáris egyenletrendszerre alkalmazott Newton módszer globálisan konvergens, sőt tetszőleges $\mathbf{x}_0$ kezdővektor esetén $\mathbf{x}_1$ az egyenletrendszer (pontos) megoldása lesz.

9.3.7. Feladat. Keressen példát olyan két ismeretlenes egyenletrendszerre, ahol $\mathbf{F}(\mathbf{x})$ mindenütt konvex függvény, az $\mathbf{F}(\mathbf{x}) = \mathbf{0}$ egyenletrendszernek egyetlen $\mathbf{x}^*$ megoldása van, és a Newton-módszer globálisan konvergens.

9.3.8. Feladat. Határozza meg, hogyan alkalmazható a Newton-módszer olyan $\mathbf{T}\mathbf{x}+\mathbf{G}(\mathbf{x})=\mathbf{0}$ speciális alakú egyenletrendszerre, ahol $\mathbf{T}$ tridiagonális mátrix, $t_{kk} = -2$, $t_{k,k+1} = t_{k+1,k} = 1$ és $\mathbf{G}$ k-dik komponensfüggvénye csak x_k-tól függ, tehát $G_k : \mathbb{R}^n \mapsto \mathbb{R}$ és $G_k(\mathbf{x}) = G_k(x_k)$.

9.3.9. Feladat. Határozza meg a következő egyenletek komplex gyökeit a valós és a komplex rész szétválasztása után kapott 2 ismeretlenes egyenletrendszer többváltozós Newton-módszerrel való megoldásával.

a) $z^3 - 1 = 0$, b) $z^4 - 1 = 0$, c) $e^z - z = 0$, d) $ze^z = 0$.

9.3.10. Feladat. Írjon a többváltozós Newton-módszert megvalósító (Maple) eljárást.
Az eljárás paraméterezése `NewtonSys(J, F, x0, kMAX, eps1, eps2)` alakú legyen, ahol

`NewtonSys`	az eljárás neve,
`J`	az $\mathbf{F}'(\mathbf{x})$ Jacobi-mátrix,
`F`	az $\mathbf{F}(\mathbf{x}) = \mathbf{0}$ egyenletrendszer bal oldalát megadó vektor,
`x0`	az $\mathbf{x}_0$ kezdeti közelítés vektora,
`kMAX`	az iterációs lépések maximális száma,
`eps1`	a k-dik közelítés megengedett első hibája: $\left\|\mathbf{F}(\mathbf{x}_k)\right\| < eps1$,
`eps2`	a k-dik közelítés megengedett második hibája: $\left\|\mathbf{x}_{k+1} - \mathbf{x}_k)\right\| < eps2$

9.3.11. Feladat. Írjon a többváltozós egyszerűsített Newton-módszert megvalósító (Maple) eljárást.
Az eljárás paraméterezése `NewtonSysS(J0, F, x0, kMAX, eps1, eps2)` alakú legyen, ahol

`NewtonSysS`	az eljárás neve,
`J0`	az $\mathbf{F}'(\mathbf{x}_0)$ Jacobi-mátrix,
`F`	az $\mathbf{F}(\mathbf{x}) = \mathbf{0}$ egyenletrendszer bal oldalát megadó vektor,
`x0`	az $\mathbf{x}_0$ kezdeti közelítés vektora,
`kMAX`	az iterációs lépések maximális száma,
`eps1`	a k-dik közelítés megengedett első hibája: $\left\|\mathbf{F}(\mathbf{x}_k)\right\| < eps1$,
`eps2`	a k-dik közelítés megengedett második hibája: $\left\|\mathbf{x}_{k+1} - \mathbf{x}_k)\right\| < eps2$

9.3.12. Feladat. Írjon a Broyden-módszert megvalósító (Maple) eljárást.
Az eljárás paraméterezése `NewtonSysB(B0, F, x0, kMAX, eps1, eps2)` alakú legyen, ahol

`NewtonSysB`	az eljárás neve,
`B0`	az első lépésben felhasznált Broyden-mátrix,
`F`	az $\mathbf{F}(\mathbf{x}) = \mathbf{0}$ egyenletrendszer bal oldalát megadó vektor,
`x0`	az $\mathbf{x}_0$ kezdeti közelítés vektora,
`kMAX`	az iterációs lépések maximális száma,
`eps1`	a k-dik közelítés megengedett első hibája: $\left\| \mathbf{F}(\mathbf{x}_k) \right\| < eps1$,
`eps2`	a k-dik közelítés megengedett második hibája: $\left\| \mathbf{x}_{k+1} - \mathbf{x}_k) \right\| < eps2$

9.3.13. Feladat. Próbálja meg a 9.3.10–9.3.12. feladatok programjaival meghatározni a fejezetben szereplő korábbi egyenletrendszerek megoldásait.

9.3.14. Feladat. Közelítse a következő egyenletrendszer megoldását a Newton-módszer három változatával, különböző kezdőértékekkel. Mit tapasztal?

$$\begin{aligned} x_1^2 + 5x_1x_2 + 22x_3 &= 0 \\ x_1 + \quad 2x_2 + 10x_3 &= 0 \\ x_1x_3 + \quad 2x_2^2 + 15x_3 &= 0 \end{aligned}$$

9.3.15. Feladat. Határozza meg függvényminimalizálással az

$$\begin{aligned} F_1(x_1, x_2) = x_1^2 + x_2 - 5 &= 0 \\ F_2(x_1, x_2) = x_1 + x_2^2 + 2 &= 0 \end{aligned}$$

egyenletrendszer megoldását.

10. fejezet

Numerikus Integrálás

10.1. Interpolációs kvadratúra-formulák

10.1.1. Feladat. Keressen az $I_n = \frac{1}{e}\int_0^1 x^n e^x\,dx$ integrálok sorozatának meghatározására alkalmas egyszerű rekurzív képletet, s számítsa ki ennek segítségével I_{20}-at.

10.1.2. Feladat. A 10.1.1. Feladatban szereplő I_{20} integrált közelítse háromféle módon:

a) közvetlenül I_{20}-at trapézszabállyal, 10^{-8} pontossággal;

b) I_0-t trapézszabállyal, 10^{-8} pontossággal, majd ebből kiindulva a rekurzív képlettel határozza meg I_{20}-at;

c) közvetlenül I_{40}-et trapézszabállyal, 10^{-8} pontossággal, majd ebből kiindulva a rekurzív képletet „visszafelé alkalmazva” határozza meg I_{20}-at.

Hasonlítsa össze a kapott közelítések pontosságát, magyarázza meg az eredményeket.

10.1.3. Feladat. Vizsgálja a 10.1.1. és a 10.1.2. Feladatokhoz hasonlóan az $I_n = \int_0^1 \frac{x^n}{x+5}\,dx$ integrálok sorozatát.

10.1.4. Feladat. Vizsgálja a 10.1.1–10.1.3. Feladathoz hasonlóan az $I_n = \int_0^{1/3} \frac{x^n}{x+1}\,dx$ integrálok sorozatát.

10.1.5. Feladat. Bizonyítsa be, hogy ha $f(x)$ monoton nő $[a,b]$-n, akkor az intervallum m egyenlő hosszúságú részre osztása után az összetett trapézszabállyal számított t_m közelítő összegek hibájára az $\left|\int_a^b f(x)\,dx - t_m\right| \le \frac{b-a}{m}\,(f(b)-f(a))$ egyenlőtlenség teljesül.

10.1.6. Feladat. Bizonyítsa be, hogy ha $f(x)$ monoton nő $[a,b]$-n, akkor az összetett középpontos szabállyal számított e_m közelítő összegek hibájára is igaz az $\left|\int_a^b f(x)\,dx - e_m\right| \le$ $\le \frac{b-a}{m}\,(f(b)-f(a))$ egyenlőtlenség.

10.1.7. Feladat. Milyen interpolációs kvadratúra-formula vezethető le a

$$\begin{aligned} H_4(a) &= f(a), \quad & H_4(b) &= f(b),\\ H_4'(a) &= f'(a), \quad & H_4'(b) &= f'(b) \end{aligned}$$

feltételekkel megadott $H_4(x)$ Hermite interpolációs polinom integrálásával? Mit állíthatunk a képlethibáról „elég sima” $f(x)$ esetében?

10.1.8. Feladat. Milyen összetett kvadratúra szabály vezethető le a 10.1.7. Feladat formulájából? Milyen kapcsolatban van ez a közelítés az összetett trapézszabállyal?

10.1.9. Feladat. Legyen $f(x) \in C^6[-1,1]$. Milyen interpolációs kvadratúra-formula vezethető le a

$$\begin{aligned} &H_6(-1)=f(-1), \quad H_6(0)=f(0), \quad H_6(1)=f(1), \\ &H_6'(-1)=f'(-1), \quad H_6'(0)=f'(0), \quad H_6'(1)=f'(1) \end{aligned}$$

feltételekkel megadott $H_6(x)$ Hermite interpolációs polinom integrálásával? Bizonyítsa be, hogy a kapott formula rendje 6. Mit állíthatunk a képlethibáról?

10.1.10. Feladat. Milyen összetett kvadratúra szabály vezethető le a 10.1.9. Feladat formulájából?

10.2. Gauss-kvadratúra

10.2.1. Feladat. Igazolja, hogy az

$$\int_{-1}^{1} f(x)\,dx \approx \frac{1}{9}\left(5f\left(-\sqrt{\frac{3}{5}}\right)+8f(0)+5f\left(\sqrt{\frac{3}{5}}\right)\right)$$

formula pontos minden ötödfokú polinomra.

10.2.2. Feladat. Közelítse a $[0,1]$ intervallumhoz tartozó 4 alappontos Legendre-Gauss formulával az $\int_0^1 \frac{\sin x}{x+1}\,dx$ integrált.

10.2.3. Feladat. Határozza meg a $[0,\infty]$ intervallumhoz és a $\rho(x)=e^{-x}$ súlyfüggvényhez tartozó 3 alappontos Gauss-Laguerre kvadratúra formulát.
Közelítse ezzel a formulával az $\int_0^\infty \frac{e^{-x}}{x+1}\,dx$ integrált

10.2.4. Feladat. Közelítse a $[0,1]$ intervallumhoz és a $\rho(x)\equiv 1$ súlyfüggvényhez tartozó 5 alappontos Gauss kvadratúrával az $\int_0^1 x^3(\sin \pi x)^2\,dx$ integrált.

10.2.5. Feladat. Közelítse a $[-1,1]$ intervallumhoz és a $\rho(x)=\sqrt{1-x^2}$ súlyfüggvényhez tartozó 3 alappontos Gauss kvadratúrával az $\int_{-1}^{1} x\,\left|\sin x)\right|\,dx$ integrált.

10.2.6. Feladat. A $Q_n(f)=\sum_{j=1}^n w_j f(x_j)$ interpolációs kvadratúra-formulát *Csebisev-típusú* formulának nevezzük, ha a súlyok mind egyenlőek: $w_j=w$, $j=1,2,\dots,n$-re. Ebből rögtön adódik $w=\frac{1}{n}\int_a^b \rho(x)\,dx$. Az $s_k=\frac{1}{w}\int_a^b x^k\rho(x)\,dx$ rövidítést bevezetve a formula alappontjait az

$$\begin{aligned} x_1+x_2+\dots+x_n &= s_1 \\ x_1^2+x_2^2+\dots+x_n^2 &= s_2 \\ &\vdots \\ x_1^n+x_2^n+\dots+x_n^n &= s_n \end{aligned}$$

egyenletrendszer $[a,b]$-be eső megoldásai adják (föltéve, hogy létezik ilyen megoldás!). Az egyenletrendszer megoldása helyett a Newton-Wahring képletek felhasználásával írjon föl olyan $p(x) = x^n + b_1 x^{n-1} + \dots + b_n$ n-ed fokú polinomot, melynek gyökei megegyeznek az eredeti egyenletrendszer megoldásával.
Vizsgálja ennek alapján az $[a,b] = [-1,1]$, $\rho(x) \equiv 1$ esetnek megfelelő Csebisev-típusú kvadratúrák létezését, tulajdonságait.

10.2.7. Feladat. Vizsgálja a 10.2.6. feladathoz hasonlóan az $[a,b]=[0,\infty]$, $\rho(x)=e^{-x}$ esetnek megfelelő Csebisev-típusú kvadratúrák létezését, tulajdonságait. Mutassa meg, hogy $n=3$-ra nem létezik ilyen formula.

10.2.8. Feladat. Mutassa meg, hogy a Gauss-kvadratúra w_j súlyai megkaphatók, ha az x_j alappontokat behelyettesíti a $p_0(x)$, $p_1(x)$, $\dots$, $p_{n-1}(x)$ ortogonális polinomokba és megoldja a

$$\begin{aligned} w_1 + w_2 + \dots + w_n &= \int_a^b p_0^2 \rho(x)\,dx \\ w_1 p_1(x_1) + w_2 p_1(x_2) + \dots + w_n p_1(x_n) &= 0 \\ &\vdots \\ w_1 p_{n-1}(x_1) + w_2 p_{n-1}(x_2) + \dots + w_n p_{n-1}(x_n) &= 0 \end{aligned}$$

lineáris egyenletrendszert.

10.2.9. Feladat. Mutassa meg, hogy a $Q_n(f) = \sum_{j=1}^n w_j f(x_j)$ Gauss kvadratúra-formulával kapcsolatban az alábbiak teljesülnek.

a) a $q_j(x) = \frac{p_n(x)}{(x-x_j)p_n'(x_j)}$ képlettel fölírt $q_1(x), q_2(x), \dots, q_n(x)$ polinomok páronként ortogonálisak;

b) $w_j = \langle q_j, g_j \rangle$ minden $1 \le j \le n$-re;

c) $x_j = \frac{\langle x q_j, g_j \rangle}{\langle q_j, g_j \rangle}$ minden $1 \le j \le n$-re.

10.3. Romberg integrálás

10.3.1. Feladat. Bizonyítsa be, hogy „elég sima” $f(x)$ esetében érvényesek a következő közelítő hibabecslések:

a) az összetett trapézformulánál: $\int_a^b f(x)\,dx - t_{2n} \approx \frac{1}{3}(t_{2n} - t_n)$;

b) az összetett Simpson-formulánál: $\int_a^b f(x)\,dx - s_{2n} \approx \frac{1}{16}(s_{2n} - s_n)$.

10.3.2. Feladat. Hogyan kapcsolódnak a Romberg integráláshoz a 10.3.1. Feladatban levezetett közelítő hibabecslések?

10.3.3. Feladat. Határozza meg az $I = \int_0^1 \frac{1}{2+\cos(\pi x)}\, dx$ integrál értékét Romberg-integrálással, hasonlítsa össze az eredményeket az összetett trapéz- és Simpson-szabállyal kapható közelítésekkel.

10.3.4. Feladat. Határozza meg az $I = \int_0^6 e^x \left|\sin(6\pi x)\right|\, dx$ integrál értékét Romberg-integrálással. Mi magyarázza a lassúbb konvergenciát?

10.3.5. Feladat. Mutassa meg, hogy ha $f(x)$ integrálható $[a, b]$-n, akkor a Romberg-mátrix bármely oszlopának elemei konvergálnak az integrálhoz, azaz tetszőleges, rögzített k esetén $\lim_{j\to\infty} T_{j,k} = \int_a^b f(x)\, dx$.

10.3.6. Feladat. Ismert, hogy az egységkörbe írt szabályos n-szögek t_n területképletéből a

$$t_n = \frac{n}{2}\sin\frac{2\pi}{n} < \pi$$

egyenlőtlenség adódik, továbbá a $\sin x$ sorfejtéséből a

$$\begin{aligned} t_n &= \frac{n}{2}\left(\frac{2\pi}{n} - \frac{1}{6}\left(\frac{2\pi}{n}\right)^3 + \frac{1}{120}\left(\frac{2\pi}{n}\right)^5 \pm \cdots\right) \\ &= \pi - \frac{\alpha_2}{n^2} + \frac{\alpha_4}{n^4} - \frac{\alpha_6}{n^6} \pm \cdots \end{aligned}$$

formulát kapjuk. Mutassa meg, hogy ha kiszámítja t_n-t az $n = 4, 8, 16, 32, \ldots$ értékekre, akkor a fönti sorfejtés alapján a Romberg-sémára emlékeztető alábbi extrapolációs képletekkel sokkal jobb közelítéseket kaphat π-re:

$$\text{legyen} \quad t_{n,0} = t_n \quad \text{és} \quad t_{n,k+1} = t_{2n,k} + \frac{t_{2n,k} - t_{n,k}}{4^{k+1} - 1}.$$

10.3.7. Feladat. (Adaptív kvadratúra) A különféle kvadratúra-eljárások ügyes kombinálásával olyan „automatikus” vagy „adaptív” integráló eljárások készíthetők, amelyek az $I = \int_a^b f(x)\, dx$ integrált az $[a, b]$ intervallum rekurzív felosztásával, az egyes részintervallumokon kapott garantált pontosságú közelítések összeadásával próbálják közelíteni.
A Romberg integrálásra alapozva közelítse az I integrált ε pontossággal a következő módon. Ha adott a $h_j = b_j - a_j$ hosszúságú $[a_j, b_j]$ részintervallum, akkor határozza meg a

$$T_{00} = T(h_j),\ T_{10} = T(h_j/2),\ T_{20} = T(h_j/4)$$

trapéz-összegeket, majd ezekből a T_{11} és T_{21} extrapolált értékeket (ezek valójában a megfelelő Simpson-összegek).
Az $I_j = T_{21}$ integrálközelítés hibáját a $|T_{21} - T_{11}|$ értékkel becsülje. Ha $|T_{21} - T_{11}| < \frac{h_j \varepsilon}{b-a}$, akkor fogadja el I_j-t.

Az ellenkező esetben ossza fel az $[a_j, b_j]$ részintervallumot két azonos hosszúságú, $[a_j, a_j + h_j/2,]$ és $[a_j + h_j/2, b_j]$ részintervallumra, és alkalmazza rekurzívan az előzőeket. Próbálja ki ezt az eljárást a fejezetben szereplő integrálok közelítésére.

10.3.8. Feladat. Írjon a Romberg algoritmust megvalósító (Maple) eljárást.
Az eljárás paraméterezése `Romberg(f, a, b, n, reszletes)` alakú legyen, ahol

`Romberg`	az eljárás neve,
`a,b`	az integrálás határai,
`n`	a Romberg-mátrix sorainak száma,
`reszletes`	kell-e az egész táblázat.

10.4. Kvadratúra-sorozatok konvergenciája

A következő feladatokban többször hivatkozunk a *Pólya–Steklov tétel* néven ismert (lásd pl. [10], 466. o.) alábbi fontos eredményre.

Állítás. *A $\{Q_n\}$ kvadratúra-sorozat akkor és csak akkor konvergens tetszőleges $f(x) \in C[a,b]$ folytonos függvényre, ha az*

a) $\{Q_n(p)\}$ *konvergens tetszőleges $p(x)$ polinomra, és*

b) $\displaystyle\sum_{i=1}^{n} \left|w_i^{(n)}\right| \le K$ *bármely $1 \le n$-re*

föltételek teljesülnek.

10.4.1. Feladat. Igazolja, hogy ha a Q_n kvadratúra-formulák sorozata minden $f(x) \in C[a,b]$ folytonos függvényre konvergens, akkor a formulák alappontjai $[a,b]$-ben mindenütt sűrűn helyezkednek el.

10.4.2. Feladat. Bizonyítsa be a pozitív kvadratúra-formulák konvergenciájára vonatkozó alábbi állítást.

Állítás. *Ha Q_n pozitív interpolációs kvadratúra-formula minden n-re, akkor a konvergenciához szükséges és elegendő csupán a Pólya-Steklov tétel a) föltétele.*

10.4.3. Feladat. Megmutatható (lásd [10]), hogy sem a zárt, sem a nyitott Newton-Cotes formulák sorozata *nem* konvergens tetszőleges $f(x) \in C[a,b]$ folytonos függvényre, mivel nem teljesítik a Pólya-Steklov tétel b) föltételét.
A fönti állítás gyakorlati alátámasztására közelítse például a Runge-függvény $\int_{-5}^{5} \frac{1}{1+x^2}\,dx$ integrálját az n-ed rendű zárt Newton-Cotes formulák felhasználásával az $n = 2,3,4,5,6,\ldots$ értékekre. Mit tapasztal?

10.4.4. Feladat. Bizonyítsa be, hogy a Gauss kvadratúra formulák sorozata minden $f(x) \in$ $\in C[a,b]$ folytonos függvényre konvergens.

10.4.5. Feladat. Támassza alá kísérleti tapasztalatokkal is a Gauss kvadratúra formulák konvergenciáját. Vizsgálja az $[a,b] = [-1,1]$ és $\rho(x) \equiv 1$ esethez tartozó Q_n Legendre-Gauss kvadratúra formulák konvergenciáját

a) „sima” függvényekre, például az $f(x) = x^2 \sin(10\pi x)$-re, illetve

b) a $g(x) = x^2 \left|\sin(10\pi x)\right|$ „fűrészfogas” változatra.

10.4.6. Feladat. Végezzen a 10.4.5. Feladathoz hasonló számításokat az $[a, b] = [-1, 1]$ és $\rho(x) = \frac{1}{\sqrt{1-x^2}}$ esetnek megfelelő Q_n Csebisev-Gauss kvadratúra formulákkal is.

10.4.7. Feladat. Az *összetett* kvadratúra szabályokra általában nem alkalmazható a Pólya-Steklov tétel, bár korábban láttunk már ezekre vonatkozó konvergencia-tételeket is például a trapéz- és a Simpson-szabály esetében.

Bizonyítsa be az alábbi állítást, melynek érdekessége, hogy „nagyon primitív”, minimális pontosságú alapformulából kiindulva is kaphatunk konvergens összetett kvadratúra szabályokat.

Állítás. *Ha a Q pozitív kvadratúraformula rendje legalább 0, akkor a belőle származtatott $\{R_m\}$ összetett kvadratúra szabályok sorozata bármely $f(x)$ integrálható függvényre konvergens.*

Itt R_m azt az összetett formulát jelenti, amikor az $[a, b]$ intervallumot m darab $\frac{b-a}{m}$ hosszúságú részre osztjuk és az egyes részintervallumokon a Q formulával kapott értékeket összeadva közelítünk.

10.4.8. Feladat. Vezesse le a 10.4.7. Feladat állításából az összetett érintő-, trapéz- és Simpson-szabály konvergenciáját.

11. fejezet

Differenciálegyenletek megoldása

11.1. Taylor-sor és fokozatos közelítések módszere

11.1.1. Feladat. Közelítse az $y' = 1 - x + y, \quad y(0) = 1$ kezdetiérték-probléma (röviden KÉP) megoldását a Taylor-sor módszerrel.

11.1.2. Feladat. Közelítse az $y' = y - \frac{2x}{y}, \quad y(0) = 1$ KÉP megoldását a Taylor-sor módszerrel.

11.1.3. Feladat. Közelítse az $y' = x + y^2, \quad y(0) = 1$ KÉP megoldását a fokozatos közelítések módszerével.

11.1.4. Feladat. Közelítse az $y' = \sqrt{x} + y^2, \quad y(0) = 0$ KÉP megoldását a fokozatos közelítések módszerével.

11.1.5. Feladat. Vizsgálja az

$$\begin{aligned} x'(t) &= ax(t) + 2y(t) \\ y'(t) &= -2x(t) + ay(t) \end{aligned}$$

$x(0) = 0, \ y(0) = 0$ paraméteres KÉP megoldását az $a \in \mathbb{R}$ paraméter különböző értékeire. Mit tapasztal? Milyen jellegűek lesznek a megoldásgörbék az $(\, x(t), \, y(t)\,), t \in [0, \, K]$ paraméteres ábrázolásnál?

11.2. Runge-Kutta módszerek és lineáris többlépéses módszerek

11.2.1. Feladat. Közelítse az $y' = e^{-y}, \quad y(0) = 0$ KÉP megoldását az $x = 0.3$ helyen

a) Euler-módszerrel, $h = 0.1$ lépésközzel.

b) a feladat $y' = z, \, z' = -y^2, \quad y(0) = 0, \, z(0) = 0$ alakú rendszerré való átírása után az erre alkalmazott Euler-módszerrel.

Mit tapasztal?

11.2.2. Feladat. Legyen

$$f(x,y)=\begin{cases}2x, & \text{ha } y\le 0;\\ 2x-\frac{4y}{x}, & \text{ha } 0<y<x^2;\\ -2x & \text{ha } y\ge x^2.\end{cases}$$

Mutassa meg, hogy az $y'=f(x,y),\ y(0)=0$ KÉP egyértelműen megoldható, bár az origo körül *nem* teljesül a Lipschitz-feltétel.
Alkalmazza a megoldás közelítésére az $y_{n+1}=y_{n-1}+2hy'_n$ „középpontszabályt” az $y_0=0$ kezdőértékkel. Hogyan viselkednek a kiszámolt közelítések?

11.2.3. Feladat. Mutassa meg, hogy ha az $y'=-\alpha y,\ y(x_0)=y_0$ KÉP megoldását valamely lineáris többlépéses módszerrel számolja, akkor a kapott közelítések felírhatók egy alkalmas homogén *differencia-egyenlet* megoldásaként.

11.2.4. Feladat. Mutassa meg, hogy az $y_{n+1}=y_n+(1-\alpha)f(x_n,y_n)+\alpha f(x_n+\frac{h}{2c},y_n+\frac{h}{2c}f(x_n,y_n))$ egylépéses módszer rendje bármely $\alpha\neq 0$-ra 2 lesz.

11.2.5. Feladat. Legyen az $y'(x)=f(x,y),\ y(x_0)=y_0$ KÉP jobb oldali $f(x,y)$ függvénye az $a\le x\le b,\ c\le y\le d$ téglán mindkét változójában monoton nemcsökkenő és folytonosan differenciálható. Mutassa meg, hogy ekkor a párhuzamosan számított

a) explicit: $\underline{y}_{n+1}=\underline{y}_n+hf(x_n,\underline{y}_n),\qquad y(x_0)=y_0$ és

b) implicit: $\overline{y}_{n+1}=\overline{y}_n+hf(x_{n+1},\overline{y}_{n+1})$

közelítések közrefogják a KÉP $Y(x)$ pontos megoldásának a megfelelő helyeken fölvett értékeit: $\underline{y}_n\le Y(a+nh)\le\overline{y}_n$, ha $0\le nh\le b-a$.
Keressen példát olyan speciális KÉP-re, amelynél jól alkalmazható a fönti eljárás.

11.2.6. Feladat. Vizsgálja az $y'(t)=1-100\pi\cos t,\ y(0)=0$ KÉP megoldását. Hasonlítsa össze a különböző közelítő módszerekkel (Taylor-, Euler-, negyedrendű Runge–Kutta-módszer) kapott megoldásokat. Mit tapasztal?

11.2.7. Feladat. Vizsgálja az $y'=-200xy^2,\ y(-1)=\frac{1}{101}$ KÉP-ra alkalmazott negyedrendű Runge-Kutta módszer kerekítési hibáját különböző h lépésközök mellett. Mit tapasztal?

11.2.8. Feladat. Közelítse az $y'=\sin(x-y),\ y(0)=\frac{\pi}{2}$ KÉP megoldását az $y_{n+1}=y_n+$ $+\frac{h}{2}\left(f(x_n,y_n)+fx_{n+1},y_{n+1})\right)$ „általánosított trapézformulával”. Milyen h-ra lesz a képlethiba 0.003-nél kisebb? Milyen h-ra konvergál a módszer?

11.2.9. Feladat. Az α paraméter mely értékeire lesz az

$$y_{n+1}=(1-\alpha)y_n+\alpha y_{n-1}+\frac{h}{12}\left((5-\alpha)y'_{n+1}+(8+8\alpha)y'_n+(5\alpha-1)y'_{n-1}\right)$$

lineáris többlépéses módszer stabil?

11.2.10. Feladat. Mennyi lesz a 11.2.9. Feladatban szereplő módszer rendje az α paraméter különböző értékeire?

11.2.11. Feladat. Vizsgálja az

$$y_{n+1} = \alpha_0 y_n + \alpha_1 y_{n-1} + \alpha_2 y_{n-2} + h(\beta_{-1} y'_{n+1} + \beta_0 y'_n + \beta_1 y'_{n-1})$$

lineáris többlépéses módszert

a) Legföljebb mennyi lehet a módszer rendje, ha megköveteli a stabilitást?

b) Határozza meg a negyedrendű módszer együtthatóit (az α_2 paraméterrel kifejezve).

c) Milyen α_2 értékekre lesz a b) szerinti módszer stabil?

11.3. Közönséges differenciálegyenletek peremérték-problémája

11.3.1. Feladat. Bizonyítsa be, hogy az $y'' = y^3$, $y(0) = 0$, $y(\alpha) = \beta$ peremérték-problémának bármely $\alpha \neq 0$-ra tetszőleges (α, β) esetén végtelen sok megoldása van.

11.3.2. Feladat. Oldja meg közelítőleg az $y''(x) - y(x) = 0$, $y(0) = 1$, $y(\frac{\pi}{2}) = 1$ peremérték-problémát a véges differenciák módszerével az $x_0 = 0$, $x_k = x_0 + kh$, $x_n = \frac{\pi}{2}$, $h = \frac{\pi}{2n}$ beosztás segítségével.

11.3.3. Feladat. Közelítse a $-y''(x) + \frac{1}{2}(y(x) + x + 1)^3 = 0$, $y(0) = 0$, $y(1) = 0$ peremérték-probléma megoldását a $[0,1]$ intervallumon a véges differenciák módszerével.

12. fejezet

Numerikus programkönyvtárak használata

12.1. A GSL és a LAPACK

A következő feladatok a GSL (Gnu Scientific Library) numerikus programkönyvtár, lásd [19], használatára vonatkoznak. Ha installálja a `libgsl.so.*` dinamikus könyvtára(ka)t, a megfelelő include fájlokat és a `gcc` fordítóprogramot, akkor a megoldásokban található C mintaprogramok Linuxon a

```
$ gcc  -o gsl-valami  -lm -lgsl gsl-valami.c
$ ./gsl-valami
```

shell parancsokkal futtathatók.

12.1.1. Feladat. A GSL numerikus programkönyvtár
`gsl_linalg_LU_decomp` és `gsl_linalg_LU_solve`
eljárását hívó C programmal számítsa ki a $\mathbf{Ax} = \mathbf{b}$ lineáris egyenletrendszer megoldását, ha

$$\mathbf{A} = \begin{pmatrix} 1 & 3 & 3 & -2 \\ 1 & 2 & -2 & -1 \\ 0 & 0 & 2 & 1 \\ -1 & -3 & 1 & -2 \end{pmatrix} \quad \text{és} \quad \mathbf{b} = \begin{pmatrix} -13 \\ -1 \\ -4 \\ 5 \end{pmatrix}.$$

12.1.2. Feladat. A GSL numerikus programkönyvtár `gsl_linalg_QR_decomp` eljárását hívó C programmal határozza meg az

$$\mathbf{A} = \begin{pmatrix} 1 & 3 & 3 & -2 \\ 1 & 2 & -2 & -1 \\ 0 & 0 & 2 & 1 \\ -1 & -3 & 1 & -2 \end{pmatrix}$$

mátrix $\mathbf{A} = \mathbf{QR}$ ortogonális-trianguláris felbontását.

12.1.3. Feladat. Közelítse a $\mathbf{H}_4$ Hilbert-mátrix sajátértékeit és sajátvektorait a GSL numerikus programkönyvtár `gsl_eigen_symmv` eljárását hívó C programmal.

12.1.4. Feladat. Közelítse periodikus köbös spline interpolációval az $f(x) = \sin x$ függvényt a $[0, 2\pi]$ intervallumban fölvett $0 < \frac{1}{2}\pi < \pi < \frac{3}{2}\pi < 2\pi$ osztáspontokkal a GSL könyvtár `gsl_interp_cspline_periodic` eljárását hívó C programmal.

12.1.5. Feladat. A GSL könyvtár `gsl_cheb_*` függvényeit hívó C program segítségével határozza meg az $f(x) = \operatorname{sgn} x$ függvény 40-ed rendű Csebisev-sorfejtését a $[-1,1]$ intervallumon.

12.1.6. Feladat. A GSL könyvtár `gsl_integration_qags` adaptív integráló eljárását hívó C program segítségével határozza meg a következő integrálokat:

a) $f(x) = \frac{\log(x)}{\sqrt{(x)}}$, $[a, b] = [0,1]$;

b) $f(x) = \sin\frac{1}{x}$, $[a, b] = [0, \frac{\pi}{2}]$;

c) $f(x) = \frac{1}{\sqrt{(1-x^2)}}$, $[a, b] = [-1,1]$.

A következő feladat a LAPACK (lásd [18]) numerikus függvénykönyvtár, pontosabban a C nyelvű CLAPACK változatának használatára vonatkozik. Ha installálja a `gcc` fordítóprogramot, a `liblapack.so.*` és a `libclapack.so.*` dinamikus könyvtárakat a megfelelő include fájlokkal együtt, akkor a megoldásokban található C mintaprogram Linuxon a

```
$ gcc  -o lapack-valami  -lm -llapack lapack-valami.c
$ ./lapack-valami
```

shell parancsokkal futtatható.

12.1.7. Feladat. Írjon olyan C programot, amely a CLAPACK könyvtár `DGESVD_` eljárásával meghatározza az

$$\mathbf{A} = \begin{pmatrix} 2.0 & 5.0 & 3.0 & 4.0 \\ 2.0 & 1.0 & 2.0 & 4.0 \\ 3.0 & 1.0 & 6.0 & 5.0 \\ 1.0 & 8.0 & 12.0 & 1.0 \end{pmatrix}$$

mátrix szinguláris értékeit.

12.1.8. Feladat. Írjon a Maple-ben olyan `define_external(...)` külső eljárás definíciót, melyet fölhasználva a CLAPACK programkönyvtár DGEEV_ eljárásával közelítheti mátrixok sajátértékeit és sajátvektorait.
Alkalmazza ezt az eljárást a következő mátrixokra:

a) $\begin{pmatrix} 2 & -2 & 0 & -1 \\ 1 & 0 & 0 & 0 \\ 0 & -1 & 2 & -2 \\ -2 & -2 & 1 & -1 \end{pmatrix}$ b) $\begin{pmatrix} 0 & 2 & -2 & 0 \\ 0 & -2 & 2 & -1 \\ -1 & 1 & 1 & 0 \\ -1 & 1 & 1 & 0 \end{pmatrix}$ c) $\begin{pmatrix} 0 & 2 & -2 & 0 \\ 0 & -2 & 2 & -1 \\ -1 & 1 & 1 & 0 \\ -1 & 1 & 1 & 2 \end{pmatrix}$

Második rész

Megoldások

13. fejezet

A numerikus analízis alapfogalmai

13.1. Egész számok ábrázolása

1.1.1. Feladat. 00111011; 10111011; 00001101; 00101010; 10001111.
1.1.2. Feladat. 00111011; 11000101; 00001101; 00001100; 11110001.
1.1.3. Feladat. Nem adható meg, mert $129 > 2^{8-1} - 1$.
1.1.4. Feladat. Legalább 9 bit kell. Ekkor 129 alakja: 010000001.
1.1.5. Feladat. Nem adható meg, mert $-45629 < -2^{16-1}$.
1.1.6. Feladat. Legalább 17 bit kell. Ekkor -45629 alakja: 10100110111000011.
1.1.7. Feladat. -51; -78; 108; 78.
1.1.8. Feladat. -61; -77; 108; 70.
1.1.9. Feladat. Megoldandó az $x + 2^m = 2^{m-1} + |x|$ egyenlet. Ebből $x = -2^{m-2}$.

13.2. Lebegőpontos számok ábrázolása

1.2.2. Feladat. Mivel -4.6 kettes számrendszerbeli kanonikus alakja $-1.00\dot{1}001\dot{1} \cdot 2^2$, ezért az előjel bit 1, a 2 kitevőhöz tartozó 8 bit: 10000001, míg a lefelé kerekítés miatt a 23 mantissza bit: 00100110011001100110011.
Így: $-4.6 = 11000000100100110011001100110011$.
A többi pedig:

$$\begin{aligned}
13.2 &= 01000001010100110011001100110011 \\
135.5 &= 01000011000001111000000000000000 \\
105.25 &= 11000010110100101000000000000000.
\end{aligned}$$

1.2.3. Feladat. Mivel az első bit 1, így a szám negatív. A $2-9$ bitek számértéke 193, így a kitevő 66. Ezért

$$11100000111010000000000000000000 \text{ értéke } -(1+2^{-1}+2^{-3}+2^{-5})2^{66}.$$

A többi pedig:

$$11000111010110000000000000000000 \text{ értéke } -(1+2^{-1}+2^{-3}+2^{-4})2^{15}$$

$$01100111001101000000000000000000 \text{ értéke } +(1+2^{-2}+2^{-3}+2^{-5})2^{79}.$$

1.2.4. Feladat. Mivel egyszeres pontosság esetén a kerekítés miatt $2^{35}+1000\ 2^{35}$-tel egyenlő, így x lebegőpontos számértéke 0 lesz.
1.2.5. Feladat. Dupla pontosság esetén nincs kerekítési hiba, ezért x lebegőpontos értéke 1000.
1.2.6. Feladat. Szükséges 1 előjel bit, minimum 3 kitevő bit és minimum 5 mantissza bit, vagyis összesen 9 bit.

13.3. Hibaanalízis

1.3.4. Feladat. $\dfrac{5}{1.01} \leq x \leq \dfrac{5}{0.99}$
1.3.5. Feladat. Nem adható.
1.3.6. Feladat. Nem adható.
1.3.7. Feladat. Nem adható.
1.3.10. Feladat. *Útmutatás.* Az abszolút hiba $\leq (\widetilde{x}+\widetilde{y})\cdot 10^{-2}+(\widetilde{x}+\widetilde{x})\cdot 10^{-2}+10^{-4}=0.1702.$
1.3.11. Feladat. *Útmutatás.* A relatív hiba $\leq \max\left(\delta_x+\delta_y+\delta_x\delta_y;\, 2\delta_x+\delta_x\delta_y\right)=2\cdot 10^{-3}+10^{-6}.$
1.3.16. Feladat. 4 a pontos tizedesjegyek száma.
1.3.17. Feladat. Például 2.7182818284 vagy 2.7182818285.
1.3.18. Feladat. *Útmutatás.* Mivel $\left|\sin\left(\dfrac{999\pi}{1000}\right)-\sin(\pi)\right|=|\cos(z)|\cdot\dfrac{\pi}{1000}$ ahol $z\in\left[\dfrac{999\pi}{1000},1\right]$, így legfeljebb $\dfrac{\pi}{1000}$ az abszolút hiba.
1.3.19. Feladat. 1.
1.3.20. Feladat. *Útmutatás.* Mivel $|\ln(x)-\ln(\widetilde{x})|\leq\dfrac{1}{x-|x-\widetilde{x}|}\cdot|x-\widetilde{x}|\leq\dfrac{1}{1-|x-\widetilde{x}|}\cdot|x-\widetilde{x}|$, így ha $|x-\widetilde{x}|\leq\dfrac{1}{1001}$, akkor teljesül a kívánalom.

14. fejezet

A lineáris algebra numerikus módszerei

Eliminációs módszerek, trianguláris felbontások, mátrixinvertálás. Vektor- és mátrixnormák, vektor- és mátrixsorozatok konvergenciája, iterációs módszerek megállási feltételei. Jacobi- és Gauss-Seidel iteráció. Lineáris egyenletrendszerek perturbációja, mátrixok kondíciószáma.

14.1. Eliminációs módszerek

2.1.1. Feladat.

a)

$$\begin{pmatrix} -2 & -3 & 0 & -3 & 11 \\ -3 & -2 & -2 & -3 & -10 \\ 3 & -3 & 7 & 2 & 66 \\ -1 & 1 & -2 & 1 & -22 \end{pmatrix} \Rightarrow \begin{pmatrix} -2 & -3 & 0 & -3 & 11 \\ 0 & \frac{5}{2} & -2 & \frac{3}{2} & -\frac{53}{2} \\ 0 & -\frac{15}{2} & 7 & -\frac{5}{2} & -\frac{165}{2} \\ 0 & \frac{5}{2} & -2 & \frac{5}{2} & -\frac{55}{2} \end{pmatrix} \Rightarrow$$

$$\begin{pmatrix} -2 & -3 & 0 & -3 & 11 \\ 0 & \frac{5}{2} & -2 & \frac{3}{2} & -\frac{53}{2} \\ 0 & 0 & 1 & 2 & 3 \\ 0 & 0 & 0 & 1 & -1 \end{pmatrix} \qquad \mathbf{x} = \begin{pmatrix} 5 \\ -6 \\ 5 \\ -1 \end{pmatrix}$$

b)

$$\mathbf{x} = \begin{pmatrix} -6 \\ 6 \\ 2 \end{pmatrix}$$

c) Főelemcsere nélkül nem oldható meg.

d)

$$\begin{pmatrix} 1 & 2 & 3 & 12 \\ 4 & 5 & 6 & 30 \\ 7 & 8 & 9 & 48 \end{pmatrix} \Rightarrow \begin{pmatrix} 1 & 2 & 3 & 12 \\ 0 & -3 & -6 & -18 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\mathbf{x} = \begin{pmatrix} c \\ 6-2c \\ c \end{pmatrix}, \qquad \text{ahol } c \in \mathbb{R}.$$

e) $\mathbf{x}=\begin{pmatrix} 1 \\ -2 \\ 3 \end{pmatrix}$ f) $\mathbf{x}=\begin{pmatrix} 0 \\ 8 \\ 0 \end{pmatrix}$ g) $\mathbf{x}=\begin{pmatrix} 1 \\ -2 \\ 3 \end{pmatrix}$ h) $\mathbf{x}=\begin{pmatrix} 1 \\ -1 \\ 0 \end{pmatrix}$

2.1.2. Feladat.

a)

$$\begin{pmatrix} -2 & -3 & 0 & -3 & 11 \\ -3 & -2 & -2 & -3 & -10 \\ 3 & -3 & 7 & 2 & 66 \\ -1 & 1 & -2 & 1 & -22 \end{pmatrix} \Rightarrow \begin{pmatrix} -2 & -3 & 0 & -3 & 11 \\ 0 & \frac{5}{2} & -2 & \frac{3}{2} & -\frac{53}{2} \\ 0 & -\frac{15}{2} & 7 & -\frac{5}{2} & -\frac{165}{2} \\ 0 & \frac{5}{2} & -2 & \frac{5}{2} & -\frac{55}{2} \end{pmatrix} \Rightarrow$$

$$\begin{pmatrix} -2 & 0 & \frac{-12}{5} & -\frac{6}{5} & -\frac{104}{5} \\ 0 & \frac{5}{2} & -2 & \frac{3}{2} & -\frac{53}{2} \\ 0 & 0 & 1 & 2 & 3 \\ 0 & 0 & 0 & 1 & -1 \end{pmatrix} \Rightarrow \begin{pmatrix} -2 & 0 & 0 & \frac{18}{5} & -\frac{68}{5} \\ 0 & \frac{5}{2} & 0 & \frac{11}{2} & -\frac{41}{2} \\ 0 & 0 & 1 & 2 & 3 \\ 0 & 0 & 0 & 1 & -1 \end{pmatrix} \Rightarrow$$

$$\begin{pmatrix} -2 & 0 & 0 & 0 & -\frac{65}{5} \\ 0 & \frac{5}{2} & 0 & 0 & -\frac{30}{2} \\ 0 & 0 & 1 & 0 & 5 \\ 0 & 0 & 0 & 1 & -1 \end{pmatrix} \qquad \mathbf{x} = \begin{pmatrix} 5 \\ -6 \\ 5 \\ -1 \end{pmatrix}$$

c) Főelemcsere nélkül nem oldható meg.

d) Nem oldható meg Gauss-Jordan eliminációval.

2.1.3.c. Feladat.

$$\begin{pmatrix} 0 & 3 & 1 & 8 \\ -2 & 10 & 2 & -24 \\ -3 & -1 & 2 & -86 \end{pmatrix} \Rightarrow \begin{pmatrix} -3 & -1 & 2 & -86 \\ -2 & 10 & 2 & -24 \\ 0 & 3 & 1 & 8 \end{pmatrix} \Rightarrow \begin{pmatrix} -3 & -1 & 2 & -86 \\ 0 & \frac{32}{3} & \frac{2}{3} & \frac{100}{3} \\ 0 & 3 & 1 & 8 \end{pmatrix} \Rightarrow$$

$$\begin{pmatrix} -3 & -1 & 2 & -86 \\ 0 & \frac{32}{3} & \frac{2}{3} & \frac{100}{3} \\ 0 & 0 & \frac{13}{16} & -\frac{11}{8} \end{pmatrix} \qquad \mathbf{x} = \begin{pmatrix} \frac{344}{13} \\ \frac{42}{13} \\ \frac{-22}{13} \end{pmatrix}.$$

2.1.4.a. Feladat.

$$\begin{pmatrix} -2 & -3 & 0 & -3 & 11 \\ -3 & -2 & -2 & -3 & -10 \\ 3 & -3 & 7 & 2 & 66 \\ -1 & 1 & -2 & 1 & -22 \end{pmatrix} \Rightarrow \begin{pmatrix} 7 & -3 & 3 & 2 & 66 \\ -2 & -2 & -3 & -3 & -10 \\ 0 & -3 & -2 & -3 & 11 \\ -2 & 1 & -1 & 1 & -22 \end{pmatrix} \Rightarrow$$

$$\begin{pmatrix} 7 & -3 & 3 & 2 & 66 \\ 0 & -\frac{20}{7} & -\frac{15}{7} & -\frac{17}{7} & \frac{62}{7} \\ 0 & -3 & -2 & -3 & 11 \\ 0 & \frac{1}{7} & -\frac{1}{7} & \frac{11}{7} & -\frac{22}{7} \end{pmatrix} \Rightarrow \begin{pmatrix} 7 & -3 & 3 & 2 & 66 \\ 0 & -3 & -2 & -3 & 11 \\ 0 & -\frac{20}{7} & -\frac{15}{7} & -\frac{17}{7} & \frac{62}{7} \\ 0 & \frac{1}{7} & -\frac{1}{7} & \frac{11}{7} & -\frac{22}{7} \end{pmatrix} \Rightarrow$$

$$\begin{pmatrix} 7 & -3 & 3 & 2 & 66 \\ 0 & -3 & -2 & -3 & 11 \\ 0 & 0 & -\frac{5}{21} & \frac{3}{7} & -\frac{34}{21} \\ 0 & 0 & -\frac{5}{21} & \frac{10}{7} & -\frac{55}{21} \end{pmatrix} \Rightarrow \begin{pmatrix} 7 & -3 & 3 & 2 & 66 \\ 0 & -3 & -2 & -3 & 11 \\ 0 & 0 & \frac{10}{7} & -\frac{5}{21} & -\frac{55}{21} \\ 0 & 0 & \frac{3}{7} & -\frac{5}{21} & -\frac{34}{21} \end{pmatrix} \Rightarrow$$

$$\begin{pmatrix} 7 & -3 & 3 & 2 & 66 \\ 0 & -3 & -2 & -3 & 11 \\ 0 & 0 & \frac{10}{7} & -\frac{5}{21} & -\frac{55}{21} \\ 0 & 0 & 0 & -\frac{1}{6} & -\frac{5}{6} \end{pmatrix} \qquad \mathbf{x} = \begin{pmatrix} 5 \\ -6 \\ 5 \\ -1 \end{pmatrix}$$

2.1.6. Feladat.

a)

$$L = \begin{pmatrix} 1 & 0 & 0 & 0 \\ \frac{3}{2} & 1 & 0 & 0 \\ -\frac{3}{2} & -3 & 1 & 0 \\ \frac{1}{2} & 1 & 0 & 1 \end{pmatrix}, \qquad U = \begin{pmatrix} -2 & -3 & 0 & -3 \\ 0 & \frac{5}{2} & -2 & \frac{3}{2} \\ 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \mathbf{x} = \begin{pmatrix} 5 \\ -6 \\ 5 \\ -1 \end{pmatrix}$$

b)

$$L = \begin{pmatrix} 1 & 0 & 0 \\ -\frac{1}{3} & 1 & 0 \\ -4 & \frac{15}{4} & 1 \end{pmatrix}, \qquad U = \begin{pmatrix} -3 & -2 & 1 \\ 0 & -\frac{8}{3} & -\frac{8}{3} \\ 0 & 0 & 13 \end{pmatrix}, \mathbf{x} = \begin{pmatrix} -6 \\ 6 \\ 2 \end{pmatrix}$$

2.1.7. Feladat.

$$L = \begin{pmatrix} 2 & 0 & 0 \\ 1 & 1 & 0 \\ \frac{1}{2} & \frac{1}{2} & \sqrt{2} \end{pmatrix}, \qquad \mathbf{x} = \begin{pmatrix} 1 \\ -1 \\ 0 \end{pmatrix}$$

2.1.8. Feladat.

a)

$$\begin{pmatrix} -\frac{15}{52} & -\frac{19}{26} & -\frac{1}{52} & -\frac{3}{26} \\ -\frac{19}{26} & -\frac{5}{13} & -\frac{3}{26} & \frac{4}{13} \\ -\frac{1}{52} & -\frac{3}{26} & -\frac{7}{52} & \frac{5}{26} \\ -\frac{3}{26} & \frac{4}{13} & \frac{5}{26} & \frac{2}{13} \end{pmatrix}$$

b)

$$\begin{pmatrix} 3 & -3 & 1 \\ -3 & 5 & -2 \\ 1 & -2 & 1 \end{pmatrix}$$

2.1.10. Feladat.

$$\begin{pmatrix} 1 & 0 & c & 5 \\ 1 & 1 & 1 & 4 \\ c & 0 & 1 & c+4 \end{pmatrix} \Rightarrow \begin{pmatrix} 1 & 0 & c & 5 \\ 0 & 1 & 1-c & -1 \\ 0 & 0 & 1-c^2 & 4-4c \end{pmatrix}$$

Ha $c = -1$, akkor nincs megoldás.

Ha $c^2 \neq 1$, akkor pontosan egy megoldás van: $\mathbf{x} = \begin{pmatrix} \frac{5+c}{1+c} \\ \frac{-5+3c}{1+c} \\ \frac{4}{1+c} \end{pmatrix}$.

Ha $c = 1$, akkor végtelen sok megoldás van: $\mathbf{x} = \begin{pmatrix} 5-d \\ -1 \\ d \end{pmatrix}$, ahol $d \in \mathbb{R}$.

2.1.13. Feladat. Igen, nem, nem.

2.1.20. Feladat. A Gauss-eliminációt megvalósító függvény a programlisták 127. oldalán található.

2.1.21. Feladat. A Gauss-Jordan-eliminációt megvalósító függvény a programlisták 128. oldalán található.

14.2. Vektor- és mátrixnormák, Jacobi és Gauss-Seidel iteráció

2.2.1. Feladat.

a) $||\mathbf{v}||_1 = 2$, $||\mathbf{v}||_2 = \sqrt{2}$, $||\mathbf{v}||_\infty = 1$.

b) $||\mathbf{v}||_1 = 6$, $||\mathbf{v}||_2 = \sqrt{14}$, $||\mathbf{v}||_\infty = 4$.

c) $||\mathbf{v}||_1 = 10$, $||\mathbf{v}||_2 = \sqrt{30}$, $||\mathbf{v}||_\infty = 4$.

2.2.6. Feladat.

a) $||A||_1 = 4$, $||A||_2 = \sqrt{\dfrac{15+\sqrt{29}}{2}}$, $||A||_\infty = 4$.

b) $||A||_1 = 10$, $||A||_2 = 10$, $||A||_\infty = 10$.

2.2.9. Feladat. 12, 11, 11.

2.2.10. Feladat. 11, 11.

2.2.15. Feladat.

a) $\mathbf{x}_4 = \left(\dfrac{33}{16}, \dfrac{17}{16}\right)^T$. b) $\mathbf{x}_3 = \left(\dfrac{24}{25}, 0\right)^T$. c) $\mathbf{x}_3 = \left(\dfrac{1}{100}, \dfrac{1}{10}, \dfrac{9}{20}\right)^T$.

d) $\mathbf{x}_4 = \left(\dfrac{597}{313}, \dfrac{311}{350}, -\dfrac{48}{875}, -\dfrac{2099}{2450}\right)^T$. e) $\mathbf{x}_5 = \left(\dfrac{969}{512}, -\dfrac{233}{1908}, -\dfrac{775}{906}\right)^T$.

2.2.16. Feladat.

a) $\mathbf{x}_3 = \left(\dfrac{33}{16}, \dfrac{63}{64}\right)^T$. b) $\mathbf{x}_3 = (1, 0)^T$. c) $\mathbf{x}_2 = \left(\dfrac{1}{10}, \dfrac{1}{5}, \dfrac{9}{20}\right)^T$.

d) $\mathbf{x}_3 = \left(\dfrac{752}{351}, \dfrac{1653}{1750}, -\dfrac{161}{4801}, -\dfrac{1349}{1334}\right)^T$. e) $\mathbf{x}_3 = \left(\dfrac{2077}{1120}, -\dfrac{663}{11200}, -\dfrac{1153}{1172}\right)^T$.

2.2.19. Feladat. A Gauss-Seidel iterációt megvalósító függvény a programlisták 129. oldalán található.

14.3. Kondíciószám, perturbáció

2.3.1. Feladat.

a) $cond_1(A) = cond_\infty(A) = 7$. b) $cond_1(A) = cond_\infty(A) = \dfrac{3}{2}$.

c) $cond_1(A) = cond_\infty(A) = 69407$. d) $cond_1(A) = 25$, $cond_\infty(A) = \dfrac{77}{3}$.

15. fejezet

Függvényközelítések

15.1. Lagrange interpoláció

3.1.1. Feladat. $det(V) = (x_0 - x_1)(x_0 - x_2)(x_1 - x_2)$.

3.1.2. Feladat. $L(x) = \dfrac{x(x+1)(x+2)}{3} = \dfrac{1}{3}x^3 + x^2 + \dfrac{2}{3}x$.

3.1.4. Feladat.
a) $L_2(x) = x^2$ b) $L_3(x) = x^3$.

3.1.6. Feladat.

$$
\begin{aligned}
p_1(n) &= n \\
p_2(n) &= \frac{n(n+1)}{2} = \frac{1}{2}n^2 + \frac{1}{2}n \\
p_3(n) &= \frac{n(n+1)(2n+1)}{6} = \frac{1}{3}n^3 + \frac{1}{2}n^2 + \frac{1}{6}n \\
p_4(n) &= \left(\frac{n(n+1)}{2}\right)^2 = \frac{1}{4}n^4 + \frac{1}{2}n^3 + \frac{1}{4}n^2
\end{aligned}
$$

3.1.7. Feladat.

$$
\begin{aligned}
L_1(x) &= (e-1)x + 1 \\
L_2(x) &= 0.8417x^2 + 0.8766x + 1 \\
L_3(x) &= 0.2786x^3 + 0.4257x^2 + 1.014x + 1
\end{aligned}
$$

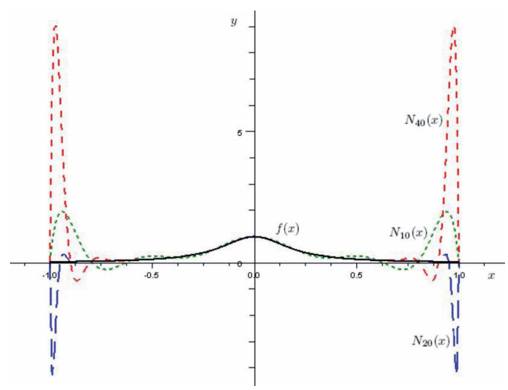
3.1.8. Feladat. *Útmutatás.* Használja az $|L_n(x) - e^x| \le \dfrac{e}{4(n+1)}\left(\dfrac{1}{n}\right)^{n+1}$ hibabecslést!

3.1.9. Feladat. *Útmutatás.* Oldja meg az $\dfrac{e}{4(n+1)}\left(\dfrac{1}{n}\right)^{n+1} \le 10^{-3}$ egyenlőtlenséget!

3.1.10. Feladat. *Útmutatás.* A $\sin(x)$ függvény speciális alakja miatt elegendő a $\left[0, \dfrac{\pi}{2}\right]$-n 10^{-3} pontossággal közelíteni. Ehhez negyedfokú polinom megfelelő.

3.1.12. Feladat. $L_4(x) = 0.028714x^4 - 0.203585x^3 + 0.019951x^2 + 0.996317x$.

3.1.13. Feladat. *Útmutatás.* Alkalmazza az interpolációt az $f(x) \equiv 1$ függvény esetében!
3.1.14. Feladat. A Lagrange interpolációt megvalósító függvény a programlisták 130. oldalán található.
3.1.18. Feladat. Az osztott differenciák sorban 0, 1, 7, 6, 1, majd innen a többi 0.
3.1.21. Feladat. Az osztott differenciák sorban 0, 0.9549, -0.2443, -0.1139.
3.1.22. Feladat.
a) $N_2(x) = x(x-1)+x$. b) $N_3(x) = x(x-1)(x-2)+3x(x-1)+x$.
3.1.23. Feladat. A Newton-interpolációt megvalósító függvény a programlisták 131. oldalán található.
3.1.25. Feladat. A 15.1 ábrán látható $n = 10$, 20 és 40 esetén az $f(x) = \dfrac{1}{1+25x^2}$ függvény $N_n(x)$ interpolációja.



15.1. ábra. Az $f(x) = \dfrac{1}{1+25x^2}$ függvény Newton interpolációs polinomjai

15.2. Hermite interpoláció

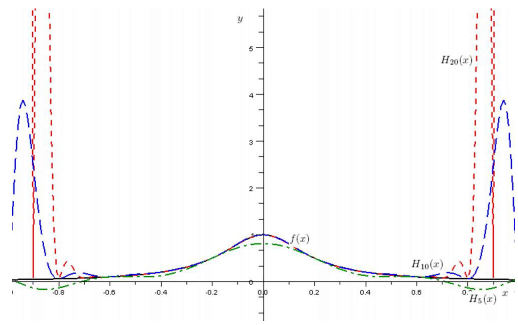
3.2.2. Feladat.
a) $H_5(x) = 1+5(x+1)-3(x+1)^2-2(x+1)^2(x-1)^2$
b) $H_3(x) = x^2+x^2(x-1)$.
3.2.3. Feladat. 4 tizedesjegyre kerekítéssel:
$H_7^{(4)}(x) = 0.9102(x-1)-0.2051(x-1)^2+0.0534(x-1)^2(x-3)-0.0055$
$(x-1)^2(x-3)^2+0.006(x-1)^2(x-3)^2(x-9)$,
8 tizedesjegyre kerekítéssel:
$H_7^{(8)}(x) = 0.9102392(x-1)-0.20511962(x-1)^2+0.05341308(x-1)^2(x-3)$
$-0.005991(x-1)^2(x-3)^2+0.0057056(x-1)^2(x-3)^2(x-9)-0.00002067$
$(x-1)^2(x-3)^2(x-9)^2+0.00000075(x-1)^2(x-3)^2(x-9)^2(x-27)$.
3.2.4. Feladat. Útmutatás: Használja a

$$|\log_3(x)-H_7(x)| \leq \frac{1}{8(z(x))^8}(x-1)^2(x-3)^2(x-9)^2(x-27)^2, \text{ ahol } x, z(x) \in [1,27]$$

egyenlőtlenséget a (kerekítési hibát nem figyelembevevő) hibabecsléshez.

3.2.5. Feladat. Az Hermite interpolációt megvalósító függvény a programlisták 132. oldalán található.

3.2.7. Feladat. A 15.2 ábrán látható $n=5$, 10 és 20 esetén az $f(x)=\dfrac{1}{1+25x^2}$ függvény $H_n(x)$ interpolációs polinomja.



15.2. ábra. Az $f(x)=\dfrac{1}{1+25x^2}$ függvény Hermite interpolációs polinomjai

3.2.8. Feladat. Útmutatás: A Hermite interpolációs polinom egyértelmű.

15.3. Spline interpoláció

3.3.4. Feladat.

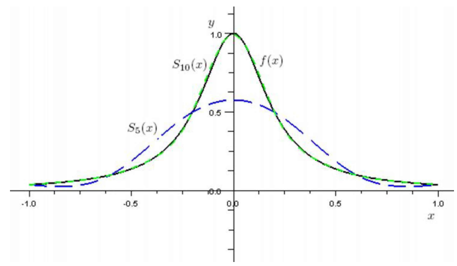
$$S(x)=\begin{cases} \dfrac{3}{2}x^3-\dfrac{1}{2}x & \text{ha } x\in[0,1]; \\ -\dfrac{3}{2}(x-1)^3+\dfrac{9}{2}(x-1)^2+4(x-1)+1 & \text{ha } x\in(1,2]. \end{cases}$$

3.3.5. Feladat.

$$S(x)=\begin{cases} \dfrac{4}{5}x^3+\dfrac{1}{5}x & \text{ha } x\in[0,1] \\ 2(x-1)^3+\dfrac{12}{5}(x-1)^2+\dfrac{13}{5}(x-1)+1 & \text{ha } x\in(1,2] \\ \dfrac{14}{5}(x-2)^3+\dfrac{42}{5}(x-2)^2+\dfrac{67}{5}(x-2)+8 & \text{ha } x\in(2,3] \end{cases}$$

3.3.6. Feladat.

$$S(x)=\begin{cases} -0.1508x^3+0.9924x & \text{ha } x\in\left[0,\dfrac{\pi}{3}\right] \\ -0.4738\left(x-\dfrac{\pi}{3}\right)^2+0.4962\left(x-\dfrac{\pi}{3}\right)+0.866 & \text{ha } x\in\left(\dfrac{\pi}{3},\dfrac{2\pi}{3}\right] \\ 0.1508\left(x-\dfrac{2\pi}{3}\right)^3-0.4738\left(x-\dfrac{2\pi}{3}\right)^2- & \\ 0.4962\left(x-\dfrac{2\pi}{3}\right)+0.866 & \text{ha } x\in\left(\dfrac{2\pi}{3},\pi\right] \end{cases}$$



15.3. ábra. Az $f(x) = \dfrac{1}{1+25x^2}$ függvény spline közelítései

3.3.12. Feladat. A 15.3 ábrán láthatók az $f(x) = \dfrac{1}{1+25x^2}$ függvény $S_n(x)$ köbös spline közelítései $n = 5$ és 10 esetén.

16. fejezet

Nemlineáris egyenletek megoldása

Intervallumfelezés, húr- és szelőmódszer, érintőmódszer, fixpont iteráció

16.1. Intervallumfelezés, húrmódszer

4.1.1. Feladat. Az $|x_n - x^*| \leq \dfrac{b-a}{2^n} \leq 10^{-2}$ egyenlőtlenségből meghatározható a szükséges n felezésszám. Az alábbi táblázatban egybefoglalva megtalálhatók az eredmények.

Feladat	Kezdő int.	Közelítő megoldás	Pontos megoldás	Lépésszám
a)	$[0,2]$	0.2578125	$0.25917110\ldots$	8
	$[2,3]$	2.5390625	$2.54264135\ldots$	7
b)	$[0,4]$	0.7109375	$0.71202172\ldots$	8
c)*	$[1,3]$	2	2	1
d)	$[-1,1]$	-0.4921875	$-0.49615847\ldots$	8
e)	$[0.1,0.2]$	0.15625	$0.15859433\ldots$	7
	$[3,4]$	3.1484375	$3.14619322\ldots$	7
f)	$[-1,0]$	-0.5703125	$-0.56714329\ldots$	7

* A c) feladatban -1 is gyök, de mivel itt -1 egyúttal lokális maximum is, így az intervallumfelezés alkalmazásához szükséges előjelfeltétel nem teljesül.

4.1.2. Feladat. Az alábbi táblázatban egybefoglalva megtalálhatók az eredmények.

Feladat	Kezdő int.	Közelítő megoldás	Lépésszám
a)	$[0,2]$	0.259725	3
	$[2,3]$	2.54203	6
b)	$[0,4]$	0.71007	67
	$[-1,1]$	0.71148	5
c)*	$[1,3]$	1.99949	10
d)	$[-1,1]$	-0.49607	2

e)	$[0.1,0.2]$	0.15907	3
	$[3,4]$	3.13844	1
f)	$[-1,0]$	-0.57218	2

* A c) feladatban -1 is gyök, de mivel itt -1 egyúttal lokális maximum is, így az húrmódszer alkalmazásához szükséges előjelfeltétel nem teljesül.

4.1.3. Feladat. Útmutatás: Bizonyítsa be, hogy az adott intervallumban a függvény végig konvex vagy konkáv és a végpontokban ellenkező előjelű.

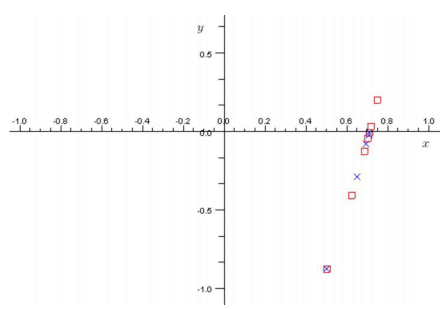
4.1.4. Feladat. Útmutatás: Használja az $|x_n - x^*| \leq \frac{|f(x_n)|}{\min_{x\in I} |f'(x)|} \leq 10^{-3}$ egyenlőtlenséget, ahol $x_n,\ x^* \in I$.

4.1.5. Feladat. $[0.5,1]$, illetve $[0.1,0.2]$ és $[3,4]$.

4.1.7. Feladat. Az intervallumfelezéses gyökközelítést megvalósító függvény a programlisták 133. oldalán található.

4.1.8. Feladat. A húrmódszert megvalósító függvény a programlisták 134. oldalán található.

4.1.11. Feladat. A 16.1 ábrán látható az intervallumfelezéses és a húrmódszer összehasonlítása a 4.1.1.b. feladat esetében. Az intervallumfelezéses módszert $\square$, a húrmódszert $\times$ jelöli.



16.1. ábra. Az intervallumfelezéses és a húrmódszer összehasonlítása

4.1.12. Feladat. $[0.5,2.5]$ egy alkalmas kezdőintervallum és n=3-ra az x_n=2.218778 megfelelő közelítés.

16.2. Érintőmódszer, szelőmódszer

4.2.1. Feladat. Az alábbi táblázatban egybefoglalva megtalálhatók az eredmények.

Feladat	Kezdőpont	Közelítő megoldás	Lépésszám
a)	1	0.25916	3
	2.5	2.54643	2
b)	2	0.71203	5
c)	-2	-1.00064	11
	2.5	2.00004	3
d)	0	-0.49616	2

e)	0.15	0.15632	1
	3.5	3.14520	2
f)	−0.5	−0.56631	1

4.2.2. Feladat. Az alábbi táblázatban egybefoglalva megtalálhatók az eredmények.

Feladat	Kezdőpont	Közelítő megoldás	Lépésszám
a)	0 és 2	0.25916	5
	2 és 3	2.54191	4
b)	0 és 1	0.71105	4
c)	−3 és −0.5	−0.99928	14
	−1.5 és −0.5	2	1
	1 és 3	2.00025	6
d)	−1 és 1	−0.49643	2
e)	0.1 és 0.2	0.15864	3
	3 és 4	3.14580	2
f)	−1 és 0	−0.56710	3

4.2.4. Feladat. Útmutatás: Az állítás következik abból, hogy a kisebbik gyök egyszeres, míg a nagyobbik kétszeres gyök.
4.2.5. Feladat. Útmutatás: Elegendő $x_0 > 0$ esetén megmutatni a konvergenciát. Ez pedig következik abból, hogy x_n, $n=1,2,\ldots$ monoton csökken és pozitív. A keresendő halmaz így: $\mathbb{R}\setminus\{0\}$.
4.2.6. Feladat. Útmutatás: $x_{n+1} = \dfrac{1}{k}\left((k-1)\,x_n + \dfrac{a}{x_n^{k-1}}\right)$.
4.2.8. Feladat. A érintőmódszert megvalósító függvény a programlisták 135. oldalán található.
4.2.9. Feladat. A szelőmódszert megvalósító függvény a programlisták 136. oldalán található.

16.3. Fixpont iteráció

4.3.2. Feladat. Az alábbi táblázatban egybefoglalva megtalálhatók az eredmények.

Feladat	x_0	Közelítő megoldás	Pontos megoldás	Lépésszám
a)	1	0.5601155	0.56714329...	7
b)	0	0.4963373	0.49615847...	2
c)	2	1.5495116	1.55714559	9
d)	−1	−0.5610121	−0.56714329...	10
e)	0.5	0.5473328	0.55714559...	4

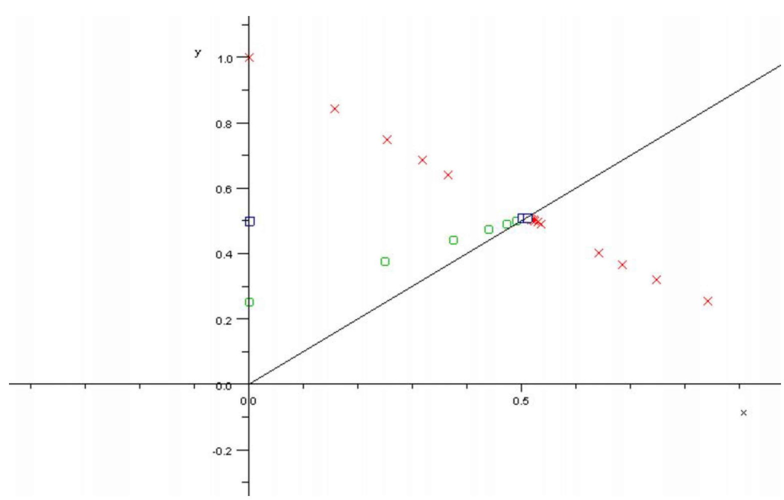
Az f) feladathoz nincs alkalmas kezdőpont.

4.3.3. Feladat.

a) $x = \dfrac{x-1-x^3}{2}$, közelítő érték: -0.6812744, $x_0 = -0.5$.
b) $x = \dfrac{x-\ln(x)}{2}$, közelítő érték: 0.5712689, $x_0 = 1$.
c) $x = \dfrac{\cos(x)}{2}$, közelítő érték: 0.4526329, $x_0 = 0.5$.

4.3.4. Feladat.
a) *Útmutatás.* a $[0,1]$-beli gyökhöz $x_{n+1} = \dfrac{e^{x_n}}{5}$, a $[2,3]$-beli gyökhöz $x_{n+1} = \ln(5x_n)$ alkalmas.
b) *Útmutatás.* a $[-1,0]$-beli gyökhöz $x_{n+1} = \dfrac{e^{x_n}-2}{2}$, az $[1,2]$-beli gyökhöz $x_{n+1} = \ln(2x_n+2)$ alkalmas.
c) *Útmutatás.* a $[-2,0]$-beli gyökhöz $x_{n+1} = \dfrac{-x_n^3+5x_n^2+8x_n-6}{8}$, az $[1,2]$-beli gyökhöz $x_{n+1} = \dfrac{x_n^3+6}{5x_n}$, a $[4,5]$-beli gyökhöz $x_{n+1} = \sqrt{5x_n - \dfrac{6}{x_n}}$ alkalmas.

4.3.5. Feladat. *Útmutatás.* Vizsgálja meg a $|g'(x)|$ értékét a gyök, $0.510973\ldots$ közelében. A 16.2 ábrán a különböző $g(x)$ függvények alapján kapott pontok sorozata látható. Az a) feladat függvényét $\times$, a b) feladatét $\square$, a c) feladatét $\circ$ jelöli.



16.2. ábra. Különböző fixpontegyenletek iterációjának az összehasonlítása

4.3.7. Feladat. A fixpont iterációt megvalósító függvény a programlisták 137. oldalán található.

4.3.9. Feladat.
a) $x = \sqrt{2+x}$ b) $x = \sqrt{2x}$ c) $x = \dfrac{1}{2+x}$

17. fejezet

Numerikus integrálás

17.1. Newton-Cotes formulák

5.1.1. Feladat. *Útmutatás.* Helyettesítse be az $f(x) \equiv 1$ függvényt!
5.1.2. Feladat. *Útmutatás.* A kapott lineáris egyenletrendszer mátrixa egy invertálható Vandermonde mátrix.
5.1.8. Feladat. $a_0 = a_1 = 1$ és $x_0 = -\frac{\sqrt{3}}{3}$, $x_1 = \frac{\sqrt{3}}{3}$.
5.1.10. Feladat. *Útmutatás.* Vegye a középpont szabály és a trapézszabály $\frac{2}{3}, \frac{1}{3}$ súlyú konvex kombinációját!
5.1.11. Feladat. *Útmutatás.* A konkávitás miatt a trapézszabály számításakor szereplő trapézok mindig a függvénygörbe alatt helyezkednek el.
5.1.12. Feladat. *Útmutatás.* Differenciálható függvény esetén a grafikonhoz húzott érintő alatt helyezkedik el a grafikon. Lásd még az 5.1.9. feladatot. Folytonos függvény pedig közelíthető differenciálható függvények sorozatával.

17.2. Összetett formulák, hibabecslés

5.2.1. Feladat. Az integrál közelítő értéke 2.05234. A pontos értéktől vett eltérése $\leq \frac{\pi^3}{24 \cdot 16} =$ $= 0.0807\ldots$. Ha 10^{-2} alá szeretnénk menni, akkor elegendő 12 osztópontot használni, mert $\frac{\pi^3}{24 \cdot 12^2} < 10^{-2}$. Ekkor az integrál közelítő értéke: 2.0057.
5.2.2. Feladat. Az alábbi táblázatban egybefoglalva megtalálhatók az eredmények.

Feladat	Integrál közelítő értéke	Integrál értéke	Osztópontok száma
a)	0.9926	1	5
b)	0.8011	0.7938...	2
c)	1.0834	1.0899...	4
d)	2.0082	2	5
e)	0.1352	0.1287...	1

f)	0.4442	$0.4388\ldots$	2

5.2.3. Feladat. A hiba $\leq \frac{M_2}{24}(b-a)h^2$ becslést használva az alábbi eredményeket kapjuk:

Feladat	h	Integrál közelítő értéke	Becsült hiba
a)	0.5	1.6289	0.0242
	0.25	1.6312	0.0060
	0.125	1.6317	0.0015
b)	0.5	−0.9544	0.0850
	0.25	−0.9885	0.0213
	0.125	−0.9971	0.0054
c)	0.5	−0.0343	0.0148
	0.25	−0.0427	0.0037
	0.125	−0.0448	0.0010

5.2.6. Feladat. Az integrál közelítő értéke 1.89612. A pontos értéktől vett eltérése $\leq \frac{\pi^3}{12\cdot 16} = 0.16149\ldots$. Ha 10^{-2} alá szeretnénk menni, akkor elegendő 17+1=18 osztópontot használni, mert $\frac{\pi^3}{12\cdot 17^2} < 10^{-2}$. Ekkor az integrál közelítő értéke: 1.9936.

5.2.7. Feladat. Az alábbi táblázatban egybefoglalva megtalálhatók az eredmények.

Feladat	Integrál közelítő értéke	Integrál értéke	Osztópontok száma
a)	−0.7120	$-0.7182\ldots$	7
b)	0.8680	$0.8749\ldots$	7
c)	3.4445	$3.4365\ldots$	13
d)	−1.4061	$-1.4142\ldots$	4
e)	0.0866	$0.0965\ldots$	2
f)	−0.4342	$-0.4388\ldots$	4

5.2.8. Feladat. A hiba $\leq \frac{M_2}{12}(b-a)h^2$ becslést használva az alábbi eredményeket kapjuk:

Feladat	h	Integrál közelítő értéke	Becsült hiba
a)	0.5	1.6375	0.0484
	0.25	1.6332	0.0121
	0.125	1.6322	0.0031
b)	0.5	−1.0918	0.1699
	0.25	−1.0231	0.0425
	0.125	−1.0058	0.0107
c)	0.5	−0.2192	0.0714
	0.25	−0.1956	0.0179
	0.125	−0.1896	0.0045

5.2.9. Feladat. Az összetett trapézformulát megvalósító függvény a programlisták 138. oldalán található.

5.2.11. Feladat. Az integrál közelítő értéke 2.00456. A pontos értéktől vett eltérése $\leq \frac{\pi^5}{2880 \cdot 16} = 0.0066\ldots$. Ha 10^{-4} alá szeretnénk menni, akkor elegendő $2 \cdot 6+1=13$ osztópontot használni, mert $\frac{\pi^5}{2880 \cdot 6^4} < 10^{-4}$. Ekkor az integrál közelítő értéke: 2.00005.

5.2.12. Feladat. Az alábbi táblázatban egybefoglalva megtalálhatók az eredmények.

Feladat	Integrál közelítő értéke	Integrál értéke	Osztópontok száma
a)	1.17560	1.1752 . . .	3
b)	2.92620	2.9253 . . .	11
c)	0.19292	0.1931 . . .	3
d)	−1.00017	−1	5
e)	−1.00013	−1	5
f)	1.57080	1.5707 . . .	5

5.2.13. Feladat. A hiba $\leq \frac{M_4}{180}(b-a)h^4$ becslést használva az alábbi eredményeket kapjuk:

Feladat	h	Integrál közelítő értéke	Becsült hiba
a)	0.5	−0.263490	0.00189
	0.25	−0.264193	0.00009
	0.125	−0.264238	0.000006
b)	0.5	0.208333	0.00834
	0.25	0.200521	0.00053
	0.125	0.200033	0.000033
c)	0.5	0.385835	0.00209
	0.25	0.386260	0.00014
	0.125	0.386292	0.000009

5.2.14. Feladat. Az összetett Simpson-formulát megvalósító függvény a programlisták 139. oldalán található.

5.2.16. Feladat. *Útmutatás.* Az összetett középpont szabály szerint kapott összeg egy integrál közelítő összeg.

5.2.17. Feladat. *Útmutatás.* Az összetett trapézszabály szerint kapott összeg egy integrál közelítő összegtől csak $\frac{\text{konstans}}{n}$-nel tér el.

5.2.18. Feladat. *Útmutatás.* Használja fel az 5.1.10 feladat állítását.

5.2.19. Feladat. *Útmutatás.* $4\int_0^1 \frac{1}{1+x^2}\,dx = \pi$ miatt elegendő $\int_0^1 \frac{4}{1+x^2}\,dx$ kellő pontosságú közelítése a középpont szabály segítségével az $x_i = \frac{2i+1}{2n}$, $i = 0, 1, \ldots, n-1$ osztópontokon.

5.2.24. Feladat. *Útmutatás.* Integrálja a $2+x+\frac{x^2}{2}+\frac{x^3}{6}+\frac{x^4}{24}+\frac{x^5}{120}$ függvényt alkalmas összetett Simpson-szabállyal.

5.2.25. Feladat. *Útmutatás.* $\pi \int_0^\pi \cos^2(x)\,dx$ integrálját határozza meg $h=\dfrac{\pi}{4}$ lépésközű összetett Simpson-szabállyal. A közelítő érték: 4.9348.

5.2.26. Feladat. *Útmutatás.* Oldja meg az $e^{|x|} - \left(\sqrt{\cos(x)}+1\right) = 0$ egyenletet pl. szelőmódszerrel 10^{-4}-es pontossággal, majd a kapott $x_1 < x_2$ gyököket felhasználva határozza meg 10^{-4}-es pontossággal az $\int_{x_1}^{x_2} \sqrt{\cos(x)}+1-e^{|x|}\,dx$ integrált pl. Simpson-szabállyal. $x_1 = -0.6396$, $x_2 = 0.6396$. A közrefogott terület közelítő értéke: 0.7229.

18. fejezet

Szélsőérték feladatok

18.1. Aranymetszés szerinti keresés

6.1.1. Feladat. Annak az intervallumnak a felezéspontját fogadjuk el közelítő megoldásnak, amelynek a hossza először kisebb, mint $2 \cdot 10^{-1}$. Induló intervallumként a feladatban megadottat tekintjük. Az eredmények az alábbi táblázatban vannak felsorolva.

Feladat	Minimumhely közelítő értéke	Lépésszám
a)	0.959	7
b)	1.5	6
c)	-0.041	7
d)	0.0344	6
e)	2.9164	7
f)	4.0122	8

6.1.2. Feladat. *Útmutatás.* Vizsgálja az $f'(x)$ függvényt az adott intervallumban.

6.1.3. Feladat. Használja az $1+\dfrac{\ln(\dfrac{2}{10^8(b-a)})}{\ln(\dfrac{\sqrt{5}-1}{2})} < n$ egyenlőtlenséget. A legkisebb n pozitív egész, ami kielégíti az egyenlőtlenséget, megfelelő lesz. Az egyes n értékek:
a.) 41, b.) 39, c.) 41, d.) 40, e.) 41, f.) 42.

6.1.4. Feladat. *Útmutatás.* A $-f(x)$ függvény minimumhelye $f(x)$ maximumhelye. Az unimodalitási kritérium miatt bontani szükséges az a), c), d) és f) esetében.

18.2. Szimplex módszer

6.2.1. Feladat. Annak az intervallumnak a felezéspontját fogadjuk el közelítő megoldásnak, amelynek a hossza először kisebb, mint $2 \cdot 10^{-1}$. Az eredmények az alábbi táblázatban vannak felsorolva.

Feladat	Minimumhely közelítő értéke	Itt a függvényérték	Lépésszám
a)	0.9375	3.0039	5
b)	3.0625	0.0031	8
c)	0.0625	0.0039	5
d)	0.0625	0.0039	5
e), f)	3.9375	-31.9768	10

6.2.2. Feladat. *Útmutatás.* Amelyik függvény esetében több minimumhely is van (pl. 6.1.1.b., 6.1.1.d. feladatok), vagy nem létezik abszolút minimum pl. 6.1.1.e. feladat, más kezdőintervallum más eredményre vezet(het).

6.2.3. Feladat. Annak a háromszögnek (illetve tetraédernek) a súlypontját fogadjuk el közelítő megoldásnak, amelynek az élei maximális hossza kisebb, mint 10^{-1}. Az eredmények az alábbi táblázatban vannak felsorolva.

Feladat	Minimumhely közelítő értéke	Itt a függvényérték	Lépésszám
a)	$(0.02083, 0.02083)$	0.000868	5
b)	$(0.02083, 0.02083)$	1.000868	5
c)	$(0.02083, 0.02083)$	1.042547	5
d)	$(1.02083, 0.52083)$	0.000434	5
e)	$(-0.5147, 0.50107, 1.01573)$	0.000681	10
f)	$(1.0174, 1.0174, 0.0243)$	0.000892	8

6.2.9. Feladat. *Útmutatás.* Indítsa el a szimplex módszert egyszer az $(1,0)$, $(0,1)$ és $(1,1)$ csúcsú kezdőszimplexszel, aztán pedig a $(-1,0)$, $(0,-1)$ és $(-1,-1)$ csúcsú szimplexszel. Az első esetben a $(1,1)$ minimumhelyhez, második esetben a $(-1,1)$ minimumhelyhez konvergál.

18.3. Gradiens módszerek

6.3.1. Feladat. Az eredmények az alábbi táblázatban találhatók.

Feladat	(x_3, y_3)	Függvényérték (x_3, y_3)-ban
a)	$(-0.0061, 0.0939)$	0.0077
b)	$(-0.0061, 0.0939)$	0.1000
c)	$(0.0553, 0.0570)$	1.1301

6.3.7. Feladat. Az optimális lépésközű gradiens módszert megvalósító függvény a programlisták 140. oldalán található.

6.3.11. Feladat. A konjugált gradiens módszert megvalósító függvény a programlisták 141. oldalán található.

19. fejezet

Ortogonális transzformációk és alkalmazásaik

19.1. Ortogonális transzformációk és ortogonális felbontások

7.1.1. Feladat.

a) $\alpha\mathbf{S} = \alpha(\mathbf{u}\mathbf{v}^T) = (\alpha\mathbf{u})\mathbf{v}^T = \mathbf{w}\mathbf{v}^T$, ahol $\mathbf{w} = \alpha\mathbf{u}$; a műveletszám $O(n)$,

b) $\mathbf{S}\mathbf{a} = (\mathbf{u}\mathbf{v}^T)\mathbf{a} = \mathbf{u}(\mathbf{v}^T\mathbf{a}) = \beta\mathbf{u}$,ahol $\beta = \mathbf{v}^T\mathbf{a}$; a műveletszám $O(n)$,

c) $\mathbf{a}^T\mathbf{S} = \mathbf{a}^T(\mathbf{u}\mathbf{v}^T) = (\mathbf{a}^T\mathbf{u})\mathbf{v}^T = \gamma\mathbf{v}^T$, ahol $\gamma = \mathbf{a}^T\mathbf{u}$; a műveletszám $O(n)$,

d) $\mathbf{S}\mathbf{A} = (\mathbf{u}\mathbf{v}^T)\mathbf{A} = \mathbf{u}(\mathbf{v}^T\mathbf{A}) = \mathbf{u}\mathbf{z}^T$,ahol $\mathbf{z}^T = \mathbf{v}^T\mathbf{A}$; a műveletszám $O(n^2)$,

e) $\mathbf{A}\mathbf{S} = \mathbf{A}(\mathbf{u}\mathbf{v}^T) = (\mathbf{A}\mathbf{u})\mathbf{v}^T = \mathbf{y}\mathbf{v}^T$,ahol $\mathbf{y} = \mathbf{A}\mathbf{u}$; a műveletszám $O(n^2)$.

7.1.2. Feladat. Az **I** egységmátrixból kiindulva alkalmazza a Sherman–Morrison formulát (lásd [17]).

7.1.5. Feladat. Ha például

$$\mathbf{A} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

akkor a sajátértékek: $1, -1, i, -i$.

7.1.6. Feladat. Nem.

7.1.12. Feladat.

a) $$\mathbf{A} = \begin{pmatrix} \frac{1}{2}\sqrt{2} & 0 & 0 & -\frac{1}{2}\sqrt{2} \\ \frac{1}{2}\sqrt{2} & 0 & 0 & \frac{1}{2}\sqrt{2} \\ 0 & 0 & -1 & 0 \\ 0 & -1 & 0 & 0 \end{pmatrix} \begin{pmatrix} 2\sqrt{2} & -\sqrt{2} & \sqrt{2} & 0 \\ 0 & 2 & -2 & 0 \\ 0 & 2 & 2 & -1 \\ 0 & 0 & 0 & 2\sqrt{2} \end{pmatrix}.$$

7.1.13. Feladat.

a) $$\mathbf{A} = \begin{pmatrix} 2/3 & 2/3 & -4/15 & -1/5 \\ 1/3 & 0 & 2/15 & 14/15 \\ 0 & 1/3 & 14/15 & -2/15 \\ -2/3 & 2/3 & -1/5 & 4/15 \end{pmatrix} \begin{pmatrix} 3 & 0 & -2/3 & 0 \\ 0 & -3 & 4/3 & -2 \\ 0 & 0 & 5/3 & -7/5 \\ 0 & 0 & 0 & 1/5 \end{pmatrix}.$$

7.1.14. Feladat.

a) $$\mathbf{A} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 2 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 2 \end{pmatrix}.$$

7.1.15. Feladat. A $\mathbf{Q}$ ortogonális mátrix $\mathbf{q}_k$ oszlopvektoraival felírható $\mathbf{a}_j = r_{j1}\mathbf{q}_1 + r_{j2}\mathbf{q}_2 + + \cdots + r_{jj}\mathbf{q}_j$, így $\mathbf{a}_j^{\mathrm{T}}\mathbf{a}_j = r_{j1}^2 + r_{j2}^2 + \cdots + r_{jj}^2$. Ekkor viszont

$$\begin{aligned} |\det \mathbf{A}| = |\det \mathbf{Q}|\, |\det \mathbf{A}| = 1 \cdot |r_{11} r_{22} \cdots r_{nn}| \le \\ \le \sqrt{r_{11}^2}\sqrt{r_{21}^2 + r_{22}^2} \cdots \sqrt{r_{n1}^2 + r_{n2}^2 \cdots r_{nn}^2} = \\ = \prod_{j=1}^{n} \sqrt{\mathbf{a}_j^{\mathrm{T}}\mathbf{a}_j} = \prod_{j=1}^{n} \|\mathbf{a}_j\|_2 . \end{aligned}$$

7.1.16. Feladat.

a) $$\mathbf{H_A} = \begin{pmatrix} -2 & \frac{1}{2\sqrt{2}} & \frac{1+\sqrt{2}}{2+\sqrt{2}} & -1 \\ \frac{2(1+\sqrt{2})}{2+\sqrt{2}} & \frac{-1}{2} & \frac{3}{2} & \frac{3(1+\sqrt{2})}{2+\sqrt{2}} \\ 0 & \frac{5}{2} & 1\frac{1}{2} & \frac{-(1+\sqrt{2})}{2+\sqrt{2}} \\ 0 & 0 & \frac{4(1+\sqrt{2})}{2+\sqrt{2}} & -1 \end{pmatrix}.$$

7.1.18. Feladat. Az elemi tükrözőmátrixok segítségével történő $\mathbf{A} = \mathbf{QR}$ ortogonális trianguláris felbontást megvalósító függvény a programlisták 142. oldalán található.

7.1.20. Feladat. Az elemi forgatómátrixok segítségével történő $\mathbf{A} = \mathbf{QR}$ ortogonális trianguláris felbontást megvalósító függvény a programlisták 143. oldalán található.

19.2. Általánosított inverz, SVD

7.2.1. Feladat.

$$\mathbf{A}^{+}=\begin{pmatrix} \frac{3}{7} & \frac{4}{7} & -\frac{1}{7} \\ -\frac{4}{7} & -\frac{3}{7} & \frac{6}{7} \\ \frac{3}{7} & -\frac{3}{7} & -\frac{1}{7} \\ \frac{1}{7} & -\frac{1}{7} & \frac{2}{7} \end{pmatrix}.$$

7.2.2. Feladat. Legyen például

$$\mathbf{A}_\varepsilon=\begin{pmatrix} 1+\varepsilon & \varepsilon \\ \varepsilon & \varepsilon \end{pmatrix}, \text{ ekkor } \mathbf{A}_\varepsilon^{+}=\mathbf{A}_\varepsilon^{-1}\begin{pmatrix} 1 & -1 \\ -1 & \frac{1+\varepsilon}{\varepsilon} \end{pmatrix}.$$

Ha $\varepsilon \to 0$,

$$\lim_{\varepsilon\to 0}\mathbf{A}_\varepsilon=\mathbf{A}_0=\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, \text{ és } \mathbf{A}_0^{+}=\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}.$$

Viszont a $\lim_{\varepsilon\to 0}\mathbf{A}_\varepsilon^{+}$ határértékmátrix nem is létezik.
7.2.3. Feladat. *Útmutató.* Beszorzásokkal ellenőrizze a négy tulajdonság teljesülését.
7.2.5. Feladat. *Útmutatás.* Helyettesítse be **A** és **B** SVD-felbontását az egyenletrendszerbe, és megfelelő átalakításokkal vezesse vissza a homogén rendszer vizsgálatára.
7.2.6. Feladat.

$$\mathbf{U}\approx\begin{pmatrix} -0.792 & 0.582 & -0.179 & -0.029 \\ -0.452 & -0.371 & 0.742 & 0.3287 \\ -0.322 & -0.510 & -0.100 & -0.791 \\ -0.252 & -0.514 & -0.638 & 0.515 \end{pmatrix},$$

$$\mathbf{S}\approx\begin{pmatrix} 1.500 \\ 0.169 \\ 0.007 \\ 0.0001 \end{pmatrix},$$

$$\mathbf{V}^{\mathrm{T}}\approx\begin{pmatrix} -0.793 & -0.452 & -0.322 & -0.252 \\ 0.582 & -0.371 & -0.510 & -0.514 \\ -0.179 & 0.742 & -0.100 & -0.638 \\ -0.029 & 0.329 & -0.791 & 0.515 \end{pmatrix}.$$

19.3. A sajátértékszámítás alapjai

7.3.5. Feladat. $\lambda_k=\beta+2\alpha\cos\frac{k\pi}{n+1}$, $k=1,2,\ldots,n$.
7.3.6. Feladat. Legyen $\mathbf{A}=\begin{pmatrix} 1 & 2i \\ 0 & 2 \end{pmatrix}$, illetve $\mathbf{B}=\begin{pmatrix} 2i & 1 \\ 1 & 0 \end{pmatrix}$.
7.3.7. Feladat. $\mathbf{R}^{\mathrm{H}}=(\mathbf{U}^{\mathrm{H}}\mathbf{A}\mathbf{U})^{\mathrm{H}}=\mathbf{U}^{\mathrm{H}}\mathbf{A}^{\mathrm{H}}\mathbf{U}=\mathbf{U}^{\mathrm{H}}\mathbf{A}\mathbf{U}=\mathbf{R}$. Ez csak úgy lehet igaz, ha **R** valós diagonális mátrix, amelynek főátlójában az **A** valós sajátértékei állnak.

7.3.8. Feladat. Legyen $\mathbf{D}_1 = \mathbf{T}^{-1}\mathbf{A}\mathbf{T}$ és $\mathbf{D}_2 = \mathbf{S}^{-1}\mathbf{B}\mathbf{S}$. A sajátértékek megegyezése miatt ekkor $\mathbf{D}_1 = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n)$ és $\mathbf{D}_2 = \mathrm{diag}(\lambda_{i_1}, \lambda_{i_2}, \ldots, \lambda_{i_n})$ írható, vagyis van olyan $\mathbf{P}$ permutációs mátrix, amellyel $\mathbf{P}^T\mathbf{D}_1\mathbf{P} = \mathbf{D}_2$. A két diagonális mátrix hasonlóságából a reláció tranzitivitása miatt már $\mathbf{A} \sim \mathbf{B}$ is következik.
7.3.12. Feladat. Az $\mathbf{I}$ egység- és a $\mathbf{O}$ zérusmátrix is szóba jöhet triviális példaként, de vannak érdekesebb példák is, például ha $\mathbf{A}$ sajátértékei mind különbözőek.
7.3.14. Feladat. Mivel az $\mathbf{A}^{\mathrm{T}}\mathbf{A}$ mátrix pozitív szemidefinit,

$$\begin{aligned}\|\mathbf{A}\|_2^2 = \rho(\mathbf{A}^{\mathrm{T}}\mathbf{A}) &\le \sum_{j=1}^{n} \lambda_j(\mathbf{A}^{\mathrm{T}}\mathbf{A}) = \|\mathbf{A}\|_F^2 \\ &\le n \max_{1\le j\le n} \lambda_j(\mathbf{A}^{\mathrm{T}}\mathbf{A}) = n\rho(\mathbf{A}^{\mathrm{T}}\mathbf{A}) = n\,\|\mathbf{A}\|_2^2\,.\end{aligned}$$

7.3.15. Feladat. Egyenlőség teljesülhet például 1 rangú mátrixokra, azonos diagonális elemeket tartalmazó diagonális mátrixokra, stb.
7.3.16. Feladat. $\mathrm{cond}_2(\mathbf{A}) \le \mathrm{cond}_F(\mathbf{A}) \le n\,\mathrm{cond}_2(\mathbf{A})$.
7.3.18. Feladat. *Útmutatás.* Mit mond a Gersgorin tétel?
7.3.20. Feladat. *Útmutatás.* Először igazolja a $\|\mathbf{QA}\| = \|\mathbf{A}\|$ és $\|\mathbf{AQ}\| = \|\mathbf{A}\|$ egyenlőségeket.

19.4. Sajátérték-számítás numerikus módszerekkel

7.4.1. Feladat. A pontos sajátértékek: $-7,\ 0,\ 4,\ 12$. Az LR-transzformáció konvergál, 20 lépés után a főátló alatti maximális abszolút értékű elem ≈ 0.000075.
7.4.2. Feladat. A pontos sajátértékek: $-8, -3, 3, 9$. Az LR-transzformáció konvergál, 13 lépés után a főátló alatti maximális abszolút értékű elem ≈ 0.000023.
7.4.3. Feladat. A pontos sajátértékek: $-8, -3, -2, -2, -1$. Az LR-transzformáció konvergál, 20 lépés után a főátló alatti maximális abszolút értékű elem ≈ 0.000082.
7.4.4. Feladat. Az algoritmus ciklizál, $\mathbf{A}_3 = \mathbf{A}_1$, de ugyanerre a mátrixra az eredeti LR transzformáció konvergál.
7.4.5. Feladat. A pontos sajátértékek: $-3, -3, 1, 9, 9$. Az LR-transzformációt alkalmazva 100 lépés után a főátló alatti maximális abszolút értékű elem $\approx 1.374\cdot 10^{48}$. A QR-transzformációnál 14 lépés után a főátló alatti maximális abszolút értékű elem ≈ 0.000088.
7.4.6. Feladat. A pontos sajátértékek: $-8, -2, -2, -2, 9$. Az LR-transzformációt alkalmazva 100 lépés után a főátló alatti maximális abszolút értékű elem $\approx 1.138\cdot 10^{65}$. A QR-transzformációnál 9 lépés után a főátló alatti maximális abszolút értékű elem ≈ 0.000046.
7.4.7. Feladat. A pontos sajátértékek: $-8, 1, 4$. Az LR-transzformáció nem alkalmazható, mivel $a_{11}=0$. A QR-transzformáció konvergál, 20 lépés után a főátló alatti maximális abszolút értékű elem ≈ 0.000065.
7.4.9. Feladat. Az LR transzformációt megvalósító függvény a programlisták 144. oldalán található.
7.4.11. Feladat. Az eltolásos QR transzformációt megvalósító eljárás a programlisták 145. oldalán található.
7.4.13. Feladat. a) $\mathbf{A}$ sajátértékei közelítőleg
$\lambda_1 \approx -8.6775042737705732$, $\lambda_2 \approx -5.4282542060729266$,

$\lambda_3 \approx -2.4839177902475438$, $\lambda_4 \approx 4.5896762700910418$.
Ezeket $\approx 10^{-9}$ hibával megkaphatjuk például a ciklikus változat kilenc menetével.
7.4.14. Feladat. Az $\hat{\mathbf{A}}$ mátrix particionált alakjából látható, hogy $\mathbf{x}_2 = (\alpha\ \mathbf{y}_2^{\mathrm{T}})^{\mathrm{T}}$ megfelelő sajátvektor lesz, ha $\alpha = \frac{1}{\lambda_2-\lambda_1}\mathbf{b}_1^{\mathrm{T}}\mathbf{y}_2$ (és természetesen $\lambda_2 \neq \lambda_1$).
7.4.16. Feladat. A három mátrix sajátértékei közelítőleg

a) $-0.4166963267, 1.211012410, 2.899620975, 4.932447788, 6.373615154,$

b) $0.08229005782, 1.131648673, 2.542638468, 4.177324322, 6.066098479$ és

c) $-0.3867464631, 0.5460081247, 2.256394662, 4.441636742, 6.142706934.$

19.5. Sajátértékek perturbációja

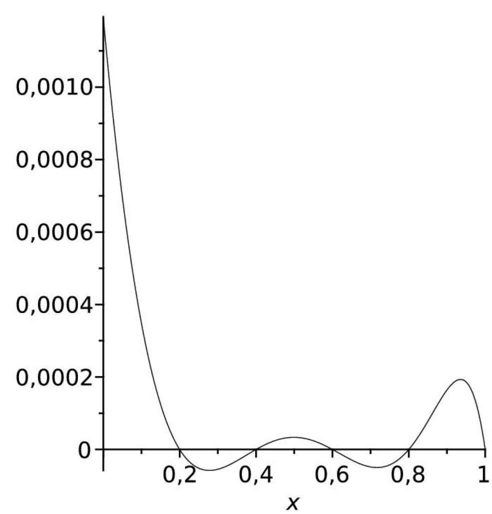
7.5.1. Feladat. *Útmutatás.* A legegyszerűbb (sőt a minimális $\|.\|_F$ normájú!) ilyen mátrix a $\hat{\Delta} = \mathbf{r}\mathbf{x}^{\mathrm{T}}$.
7.5.2. Feladat. Ha $\varepsilon \neq 0$, akkor $\lambda_1 = \lambda_2 = 1$, $\mathbf{v}_1^T = (1,0)$, $m_1 = 2$, $n_1 = 1$.
Az $\varepsilon = 0$ esetben viszont $\lambda_1 = \lambda_2 = 1$, válaszható például $\mathbf{v}_1^T = (1,0)$, $\mathbf{v}_2^T = (0,1)$, és ekkor $m_1 = n_1 = 2$.
7.5.3. Feladat. Ha $\varepsilon \neq 0$, akkor $\lambda_1 = +\sqrt{\varepsilon}$, $\lambda_2 = -\sqrt{\varepsilon}$, $\mathbf{v}_1^T = (\sqrt{\varepsilon},1)$, $\mathbf{v}_2^T = (-\sqrt{\varepsilon},1)$, $m_1 = m_2 = 1$, $n_1 = n_2 = 1$.
Az $\varepsilon = 0$ esetben viszont $\lambda_1 = \lambda_2 = 0$, $\mathbf{v}_1^T = (0,1)$, és ekkor $m_1 = 2$, $n_1 = 1$. Tehát itt $\|\Delta\mathbf{B}\| = \left\|\mathbf{B}(\varepsilon) - \mathbf{B}(0)\right\| = O(\varepsilon)$, de $|\Delta\lambda_i| = O(\sqrt{\varepsilon})$!

20. fejezet

Közelítések lineáris terekben

20.1. Interpolációs közelítések

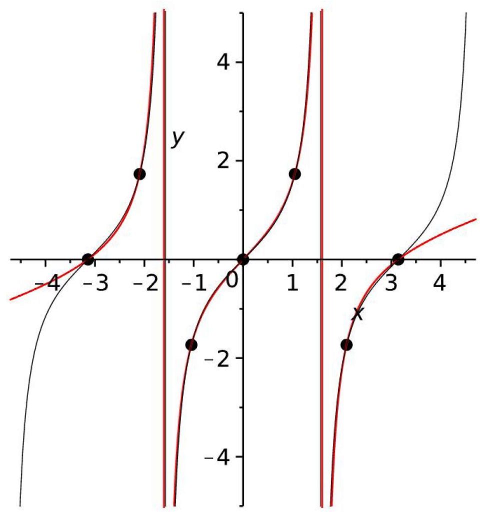
8.1.5. Feladat. Az ortogonális polinomrendszer definíciójából következik.
8.1.6. Feladat. Valójában a felsorolt alappontokhoz tartozó megfelelő Lagrange-polinomot kapjuk, hiszen $\mathcal{G}$ az összes legfeljebb n-ed fokú polinomot tartalmazza.
8.1.7. Feladat. A kapott általánosított interpolációs polinom közelítőleg $p_5(x) \approx -0.44374938 - 0.24993187e^x + 0.8287267e^{2x} - 0.149598364e^{3x} + 0.013230606e^{4x}$. A közelítés hibája, az $f(x) - p_5(x)$ eltérés látható a 20.1. ábrán.



20.1. ábra. Az általánosított interpoláció hibája

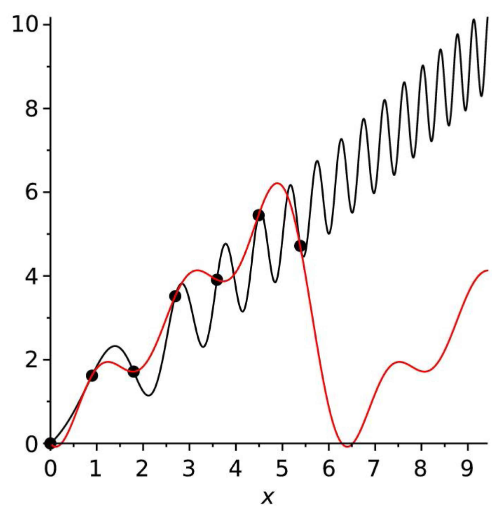
8.1.8. Feladat. Az általánosított interpolációt megvalósító eljárás a programlisták 146. oldalán található.
8.1.11. Feladat. Az $r_{1,2}(x) = \dfrac{x+1}{x^2-2x+3}$ racionális törtfüggvény az egyetlen megoldás.
8.1.12. Feladat. $r_{2,1}(x) = \dfrac{x^2-2x+3}{x+1}$, a 8.1.11. Feladat megoldásának reciproka.

8.1.13. Feladat. *Útmutatás.* Milyen megoldásai vannak a társított homogén lineáris egyenletrendszernek?
8.1.16. Feladat. A pontos megoldás $r(x)=\frac{-27\sqrt{3}x^3+27\pi^2\sqrt{3}x}{-54\pi x^2+14\pi^3}$. A 20.2. ábrán látható, hogy $r(x)$



20.2. ábra. A $\mathrm{tg}(x)$ függvény közelítése racionális interpolációval

igen jól közelít a $[-\pi,\pi]$ intervallumon, de azon kívül már nem.
8.1.17. Feladat. A $t(x)\approx 2.989557-1.238036\cos(x)-1.781577\sin(x)-0.924157\cos(2x)-$ $-0.312618\sin(2x)-0.827363\cos(3x)+0.340753\sin(3x)$ megoldás, amely a 20.3. ábrán látható, azt sugallja, hogy $f(x)$ csak igen rosszul közelíthető trigonometrikus interpolációval.



20.3. ábra. Az $f(x)$ függvény közelítése trigonometrikus interpolációval

Javulhat az eredmény, ha a közelítésben magasabb frekvenciájú tagokat is felhasználunk.

20.2. Legjobb közelítések lineáris terekben

8.2.2. Feladat. *Útmutatás.* Tekintse például az $f(x) = 1$ és a $h(x) = x$ függvényeket.
8.2.4. Feladat. *Útmutatás.* Gondoljon a 8.2.2. Feladatra és a 8.5. alfejezetben tárgyalt eredményekre.
8.2.5. Feladat. *Útmutatás.* Az egyik irányban triviális a kapcsolat (legyen $\alpha = \frac{1}{2}$). A fordított következtetéshez:

$$\left\|\alpha\mathbf{f}+(1-\alpha)\mathbf{h}\right\| \le \|\alpha\mathbf{f}\|+\left\|(1-\alpha)\mathbf{h}\right\| = \alpha(\|\mathbf{f}\|-\|\mathbf{h}\|)+\|\mathbf{h}\| = \alpha 0+\|\mathbf{h}\| = 1.$$

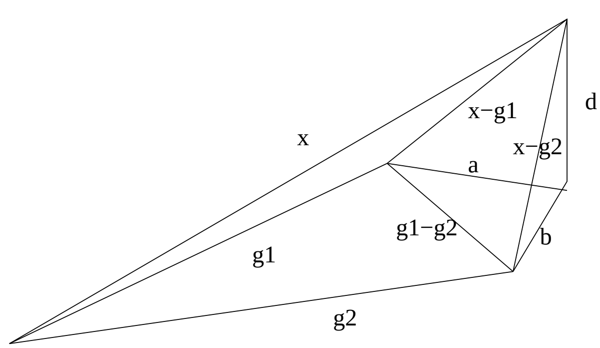
8.2.6. Feladat. Lásd [11].

20.3. Négyzetesen legjobb közelítések lineáris terekben

8.3.1. Feladat. Legyen például $[a,b] = [-1,1]$, vizsgálja a következő $\{f_n(x)\}$ függvénysorozatot:

$$f_n(x) = \begin{cases} -1 & \text{ha } x \in \left[-1, -\frac{1}{n}\right]; \\ nx & \text{ha } x \in \left[-\frac{1}{n}, \frac{1}{n}\right]; \\ +1 & \text{ha } x \in \left[\frac{1}{n}, 1\right]. \end{cases}$$

Az $f(x) = \operatorname{sgn} x$ határérték-függvény nem folytonos a 0-ban, ezért $f(x) \notin \mathsf{C}[a,b]$.
8.3.4. Feladat. *Útmutatás.* Az $\mathbf{x}$ merőleges vetületét véve a $\mathcal{G}$ síkra írja föl a Pitagorasz-tételt az $\mathbf{x}-\mathbf{g}_1$ és a $\mathbf{x}-\mathbf{g}_2$ átfogójú háromszögekre, majd alkalmazza a háromszög-egyenlőtlenséget.



20.4. ábra. A Beppo Levi egyenlőtlenség geometriai jelentése

8.3.5. Feladat. Tudjuk, hogy valós terek esetében $\mathbf{G}$ szimmetrikus. Így bármely $\mathbf{x} \in \mathbb{R}^n$-re teljesül

$$\mathbf{x} \neq \mathbf{0} \Longleftrightarrow \sum_{i=1}^{n} x_i \mathbf{g}_i \neq \mathbf{0} \in \mathcal{G} \Longleftrightarrow \left\|\sum_{i=1}^{n} x_i \mathbf{g}_i\right\|^2 > 0$$
$$\Longleftrightarrow \left\langle \sum_{i=1}^{n} x_i \mathbf{g}_i, \sum_{i=1}^{n} x_i \mathbf{g}_i \right\rangle > 0 \Longleftrightarrow \mathbf{x}^{\mathrm{T}}\mathbf{G}\mathbf{x} > 0$$

tehát **G** valóban pozitív definit.
8.3.7. Feladat. $p^*{}_2(x) \approx 0.996+1.104x+0.537x^2$.

20.4. Ortogonális polinomrendszerek

8.4.1. Feladat. $\beta = -\frac{5}{3}$.
8.4.4. Feladat. A $\mathbf{T}_k$ tridiagonális mátrix karakterisztikus polinomja pontosan a $p_k(x)$ k-dik ortogonális polinom lesz, tehát $p_k(x)$ gyökei megkaphatók valamely numerikus sajátérték-számító algorimus segítségével.
8.4.5. Feladat. *Útmutatás.* Induljon ki a 8.4.2. Feladatban megadott rekurzív képletből, használjon k szerinti indukciót.
8.4.6. Feladat. Az ortogonális polinomokat előállító eljárás a programlisták 147. oldalán található.
8.4.7. Feladat. A három tagú rekurziós képlettel számoló eljárás a programlisták 148. oldalán található.
8.4.11. Feladat. A 10.4.1. Feladat szerint így a Gauss kvadratúrák alappontjai mindenütt sűrűn helyezkednek el $[a,b]$-ben, de ezek az alappontok éppen a megfelelő ortogonális polinomok gyökei.
8.4.16. Feladat. A Maple `numapprox` csomagjának `chebisev` eljárása szerint a megfelelő Csebisev-sorfejtések:

a) $\sin x \approx 8801011715\, T_1(x) - 0.03912670797\, T_3(x) + 0.0004995154604\, T_5(x)$,

b) $\cos x \approx 0.7651976865\, T_0(x) - 0.2298069699\, T_2(x) + 0.004953277928\, T_4(x)$,

c) $e^x \approx 1.266065878\, T_0(x) + 1.130318208\, T_1(x) + 0.2714953396\, T_2(x) +$
$+0.04433684985\, T_3(x) + 0.005474240442\, T_4(x) + 0.0005429263119\, T_5(x)$.

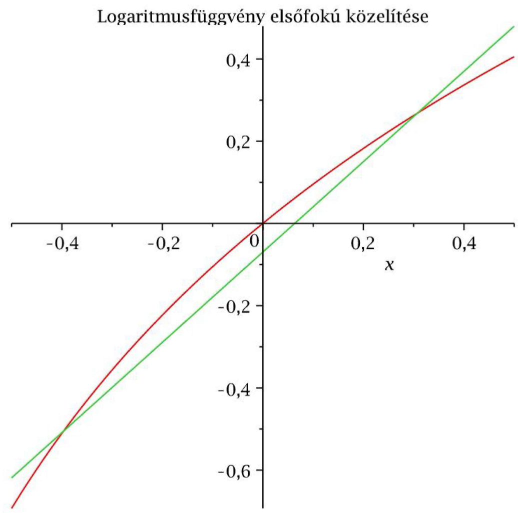
8.4.19. Feladat. *Útmutatás.* Használja az ortogonális polinomoknak a 8.4.2. Feladat b) részében megadott rekurzív képletét.

20.5. Egyenletes közelítések

8.5.1. Feladat. Csebisev tétele szerint az

$$\frac{1}{2}\left(\min_{x\in[a,b]} f(x) + \max_{x\in[a,b]} f(x)\right)$$

konstans függvény, ugyanis ennek van két optimális alternáló pontja: a és b.
8.5.2. Feladat. *Útmutatás.* Az intervallum két végpontja, a és b mindig optimális alternáló pont lesz.
8.5.3. Feladat. *Útmutatás* Mikor lehet a legjobb közelítés a $p_n(x) \equiv 0$ polinom? Csebisev tételéből induljon ki; keressen $n+2$ elemű optimális alternáló pontsorozatot.
8.5.6. Feladat. Csebisev tétele szerint elég három optimális alternáló pontot megadni. Az a és a b mellett harmadik pontként vehetjük a kiszámolt érintési pont abszcisszáját.



20.5. ábra. Az $f(x)$ függvényt egyenletesen legjobban közelítő egyenes

8.5.8. Feladat. Az $\ln(x+1) \approx -0.6963828761e-1+1.098612289\,x$ legjobban közelítő egyenes a 20.5. ábrán látható.

8.5.9. Feladat. Az állítást indirekt úton igazolhatjuk. Tegyük fel, hogy a $\mathcal{G}$ Haar-altérben az $f(x)$ függvénynek létezik két különböző legjobb közelítése, $p^*(x)$ és $p^{**}(x)$. Ismert, hogy ekkor ezek bármely konvex kombinációja is legjobb közelítés, például $q^*(x) = \frac{1}{2}(p^*(x) + p^{**}(x)) \in \mathcal{G}$ is optimálisan közelít. A $q^*(x)$ egy optimális alternáló sorozata legyen $a \le x_1 < x_2 < \cdots < x_n < x_{n+1} \le b$. Ekkor az $i = 1, 2, \ldots, n+1$ értékekre

$$f(x_i) - q^*(x_i) = \frac{1}{2}(f(x_i) - p^*(x_i)) + \frac{1}{2}(f(x_i) - p^{**}(x_i)) = \varepsilon(-1)^i E_{\mathcal{G}}(f),$$

ahol $\varepsilon = 1$ vagy -1. De mivel $\left|f(x_i) - p^*(x_i)\right| \le E_{\mathcal{G}}(f)$ és $\left|f(x_i) - p^*(x_i)\right| \le E_{\mathcal{G}}(f)$ minden pontban, egyenlőség csak úgy lehetséges, ha $f(x_i) - p^*(x_i) = f(x_i) - p^{**}(x_i)$, vagyis $p^*(x_i) = p^{**}(x_i)$. Ha viszont a $p^*(x) - p^{**}(x) \in \mathcal{G}$ általánosított polinomnak $n+1$ zérushelye van és $\mathcal{G}$ Haar-altér, akkor csak $p^*(x) \equiv p^{**}(x)$ lehet, ami ellentmondás.

8.5.11. Feladat. $\cos x \approx 0.9719959453 - 0.4052847346\,x^2$

8.5.12. Feladat. $e^{x^2} \approx 0.8941125702 + 1.718281828\,x^2$

8.5.13. Feladat. A $p_n^*(x) = \sum_{i=0}^{n} p_n^*(x_i) L_i(x)$ és az $f(x) - P_n(x) = f(x) - p_n^*(x) + p_n^*(x) -$

$-P_n(x)$ azonosságok segítségével tetszőleges $\bar{x} \in [a, b]$-re fölírható

$$\begin{aligned}
\left|f(\bar{x})-P_n(\bar{x})\right| &\leq \left|f(\bar{x})-p_n^*(\bar{x})\right|+\left|\sum_{i=0}^{n} p_n^*(x_i)L_i(\bar{x})-\sum_{i=0}^{n} f(x_i)L_i(\bar{x})\right| \\
&\leq \left|f(\bar{x})-p_n^*(\bar{x})\right|+\sum_{i=0}^{n}\left|p_n^*(x_i)-f(x_i)\right|\left|L_i(\bar{x})\right| \\
&\leq E_n(f)+E_n(f)\sum_{i=0}^{n}\left|L_i(\bar{x})\right| \\
&\leq E_n(f)\,(1+\mathcal{L}_n)\,.
\end{aligned}$$

Mivel $\bar{x} \in [a, b]$ tetszőleges volt, innen már következik a bizonyítandó egyenlőtlenség.
8.5.14. Feladat. $\max_{x\in[a,b]}\left|f(x)-P_n(x)\right| \leq (1+3.5)E_n(f)$.
8.5.15. Feladat. Induljon ki az

$$\int_a^b \left|f(x)-q_n^*(x)\right|^2 \rho(x)\,dx \leq \int_a^b \left|f(x)-p_n^*(x)\right|^2 \rho(x)\,dx$$

egyenlőtlenségből.
8.5.16. Feladat.

a) Legendre-polinomok: $\delta_n^2 \leq 2\,(E_n(f))^2$

b) Csebisev-polinomok: $\delta_n^2 \leq \pi\,(E_n(f))^2$

8.5.17. Feladat. *Útmutatás.* Az egyenlőtlenség jobb oldalának igazolásához tekintse az n darab Csebisev-alapponthoz tartozó interpolációs polinomot, a bal oldalnál induljon ki abból, hogy $p_{n-1}^*(x)$ is tekinthető interpolációs polinomnak.

21. fejezet

Egyenletrendszerek megoldása iterációs módszerekkel

21.1. Relaxációs és egyéb módszerek lineáris egyenletrendszerekre

9.1.1. Feladat. *Útmutatás.* Az

$$\left\|\mathbf{x}_{k+1}-\mathbf{x}^*\right\|_\infty = \left\|(\mathbf{I}-\mathbf{A})\mathbf{x}_k+\mathbf{b}-(\mathbf{I}-\mathbf{A})\mathbf{x}^*-\mathbf{b}\right\|_\infty = \left\|(\mathbf{I}-\mathbf{A})(\mathbf{x}_k-\mathbf{x}^*)\right\|_\infty \leq$$
$$\leq \|\mathbf{I}-\mathbf{A}\|_\infty \left\|\mathbf{x}_k-\mathbf{x}^*\right\|_\infty$$

egyenlőtlenség alapján bizonyítható a konvergencia.

9.1.2. Feladat. Ismert, hogy az $\left(\mathbf{I}-\frac{2}{\alpha}\mathbf{A}\right)$ mátrix σ_k sajátértékei kifejezhetők az $\mathbf{A}$ mátrix λ_k sajátértékeivel: $\sigma_k = 1-\frac{2}{\alpha}\lambda_k$. Tehát elegendő megmutatni, hogy $\left|1-\frac{2}{\alpha}\lambda_k\right| < 1$, azaz

$$-1 < 1-\frac{2}{\alpha}\lambda_k < 1$$

ami már könnyen belátható a λ_k sajátértékek pozitivitása és $\lambda_k < \alpha$ miatt.

9.1.3. Feladat. Az aritmetikai műveletek száma $O(n^2)$. Az iteráció fölírható

$$\mathbf{x}_{k+1} = (\mathbf{I}-\hat{\mathbf{A}}^{-1}\mathbf{A})\mathbf{x}_k+\hat{\mathbf{A}}^{-1}\mathbf{b}$$

alakban, így a konvergencia elegendő feltétele lehet $\left\|\mathbf{I}-\hat{\mathbf{A}}^{-1}\mathbf{A}\right\| < 1$ valamely alkalmasan választott mátrixnormában.

9.1.5. Feladat. *Útmutatás.* Mivel az extrapolált módszer iterációs mátrixa $\mathbf{B}_\tau = \tau\mathbf{B}+(1-\tau)\mathbf{I}$, sajátértékei $\mu_j = \tau\lambda_j+(1-\tau)$ alakúak és $-\infty < \mu_1 \leq \mu_2 \leq \cdots \leq \mu_n$. Ezért $\rho(\mathbf{B}_\tau) = \max(|\mu_1|, |\mu_n|)$.

9.1.6. Feladat. Az extrapolált Jacobi iteráció megegyezik a JOR-iterációval - csak ott τ helyett ω a a paraméter szokásos jelölése.

9.1.7. Feladat. *Útmutatás.* Elegendő például $q = \frac{\|\mathbf{B}_2\|}{1-\|\mathbf{B}\|} < \frac{1}{2}$, ekkor a közelítések hibájára belátható $\|\mathbf{e}_{k+1}\| \leq \frac{1}{2}\|\mathbf{e}_k\|$, amiből következik a konvergencia

9.1.8. Feladat. *Útmutatás.* Legyen $\mathbf{B} = \mathbf{B}_1 + \mathbf{B}_2 = \hat{\mathbf{U}} + \hat{\mathbf{L}}$, ahol $\hat{\mathbf{U}}$ és $\hat{\mathbf{L}}$ az együttható-mátrix $\mathbf{A} = \mathbf{D} - \mathbf{L} - \mathbf{U}$ felbontásából származtatott két mátrix.
9.1.9. Feladat. Ismert, hogy $\mathbf{B}_{JOR(\omega)} = \omega\mathbf{B}_J + (1-\omega\mathbf{I})$ alakban is fölírható a JOR-iteráció iterációs mátrixa. Ekkor $\mathbf{B}_{JOR(\omega)}$ μ_k sajátértékei megadhatók a $\mathbf{B}_J$ mátrix λ_k sajátértékeivel: $\mu_k = \omega\lambda_k + (1-\omega)$. Így

$$|\mu_k| = \left|\omega\lambda_k + (1-\omega)\right| \le |\omega\lambda_k| + |1-\omega| = \omega\,|\lambda_k| + 1 - \omega < \omega \cdot 1 + 1 - \omega = 1,$$

tehát a JOR-iteráció valóban konvergálni fog.
9.1.12. Feladat.

$$\mathbf{B}_{JOR(\omega)} = \begin{pmatrix} 1-\omega & \frac{1}{4} & 0 \\ \frac{1}{4} & 1-\omega & \frac{1}{4} \\ 0 & \frac{1}{4} & 1-\omega \end{pmatrix},$$

ahonnan a sajátértékek: $\lambda_1 = 1 - \omega - \frac{\sqrt{\omega}}{8}$, $\lambda_2 = 1-\omega$, $\lambda_3 = 1 - \omega + \frac{\sqrt{\omega}}{8}$. Grafikus ábrázolással látható, hogy

$$\rho(\mathbf{B}_{JOR(\omega)}) < 1 \Longleftrightarrow 0 < \omega < \frac{2\sqrt{8}}{\sqrt{8}+1}.$$

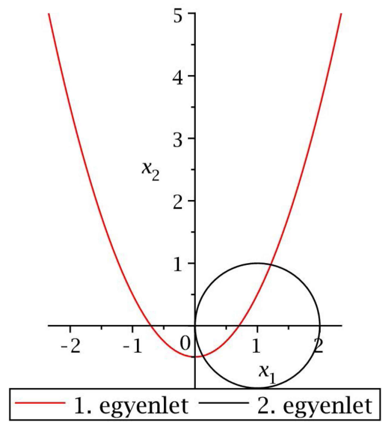
9.1.13. Feladat. *Útmutatás.* Az $\mathbf{A}$ mátrix gyengén diagonális domináns, irreducibilis, sőt pozitív definit is.
9.1.14. Feladat. *Útmutatás.* Az $\mathbf{A}$ mátrix pozitív definit lesz, ha $\frac{-1}{2} < \alpha < 1$. Ekkor az SOR iteráció bármely $0 < \omega < 2$-re konvergálni fog.
9.1.16. Feladat. A JOR módszert megvalósító Maple eljárás a programlisták 149. oldalán található.
9.1.18. Feladat. Az SOR módszert megvalósító Maple eljárás a programlisták 150. oldalán található.

21.2. Fixpontiteráció

9.2.4. Feladat. *Útmutatás.* A feltételből következik, hogy a $\mathbf{G}'(\mathbf{x})$ Jacobi-mátrix $\|.\|_1$ normája 1-nél kisebb lesz.
9.2.5. Feladat. Vizsgálja a 9.2.4. Feladat szerint, hogy az egységnégyzeten belül mikor várható konvergencia. Legyen például $\mathbf{x}_0 = (\frac{1}{2}, \frac{1}{2})^{\mathrm{T}}$.
9.2.6. Feladat. $\mathbf{x}^* \approx (-0.1605, 0.4931)^{\mathrm{T}}$.
9.2.7. Feladat. $\mathbf{x}^* \approx (1.367, -0.520)^{\mathrm{T}}$.
9.2.8. Feladat. A pontos megoldás az $\mathbf{x}^* = (\frac{1}{2}, \pi)^{\mathrm{T}}$.
9.2.9. Feladat. A 21.1. ábrán látható grafikon szerint két valós megoldás van.
9.2.10. Feladat. $\mathbf{x}^* \approx (-0.509, 0.136, 0.139)^{\mathrm{T}}$
9.2.12. Feladat. A fixpontiterációt megvalósító eljárás a programlisták 151. oldalán található.
9.2.14. Feladat. Ha $\mathbf{G}(\mathbf{x})$ κ-kontrakció az $\mathbf{x}^*$ valamely környezetében, akkor ott

$$\|\mathbf{e}_{k+1}\| = \left\|\mathbf{x}_{k+1} - \mathbf{x}^*\right\| = \left\|\mathbf{G}(\mathbf{x}_k) - \mathbf{G}(\mathbf{x}^*)\right\| \le \kappa\,\left\|\mathbf{x}_k - \mathbf{x}^*\right\| = \kappa\,\|\mathbf{e}_k\|$$

9.2.16. Feladat. *Útmutatás.* Alkalmazza a tetszőleges $\mathbf{M}$ mátrixra igaz $\rho(\mathbf{M}) \le \|\mathbf{M}\|$ egyenlőtlenséget.



21.1. ábra. Egyenletrendszer grafikus megoldása

9.2.18. Feladat. *Útmutatás.* Az $\alpha_k \to \alpha$ konvergencia miatt van olyan m küszöbszám és $0 < < \bar{\alpha} < 1$, hogy ha $k > m$, akkor $\|\mathbf{x}_{k+1} - \mathbf{x}^*\| \leq \bar{\alpha}\, \|\mathbf{x}_k - \mathbf{x}^*\|$. Ebből az egyenlőtlenségből már könnyen levezethető, hogy az $\{\mathbf{x}_k\}$ sorozat Cauchy-sorozat $\mathbb{R}^n$-ben, s így konvergens is.

21.3. A Newton-módszer általánosításai

9.3.1. Feladat. Az egyenletrendszer egyetlen megoldása $\mathbf{x}^* = (-3, 2, -4)^{\mathrm{T}}$. A Newton módszer ide konvergál például az $\mathbf{x}_0 = (1, 1, 1)^{\mathrm{T}}$ kezdővektorral.
9.3.6. Feladat. $\mathbf{F}'(\mathbf{x}) = \mathbf{A}$ miatt az iterációs képlet szerinti $\mathbf{F}'(\mathbf{x}_0)(\mathbf{x}_1 - \mathbf{x}_0) = -\mathbf{F}(\mathbf{x}_0)$ egyenletrendszer megoldása azzal ekvivalens, hogy $\mathbf{x}_1$ az $\mathbf{A}\mathbf{x}_1 = \mathbf{b}$ megoldása, tehát kielégíti az eredeti egyenletrendszert.
9.3.10. Feladat. A Newton-módszert megvalósító eljárás a programlisták 152. oldalán található.
9.3.11. Feladat. Az egyszerűsített Newton-módszert megvalósító eljárás a programlisták 153. oldalán található.
9.3.12. Feladat. A Broyden-módszert megvalósító eljárás kódja a programlisták 154. oldalán található.
9.3.14. Feladat. A triviális mellett megoldás az $\mathbf{x}^* = (1, 2, -\frac{1}{2})^{\mathrm{T}}$ is. Az $\mathbf{x}_0 = (1, 1, 1)^{\mathrm{T}}$ kezdővektorral ide konvergálnak a közelítések. Más kezdővektorra, például az $\mathbf{x}_0 = (-10, 1, -10)^{\mathrm{T}}$-re a triviális $\mathbf{x}^* = (0, 0, 0)^{\mathrm{T}}$ megoldást kapjuk, vagy nem is konvergálnak a közelítések.
9.3.15. Feladat. Minimalizálja a $G(x_1, x_2) = (x_1^2 + x_2 - 5)^2 + (x_1 + x_2^2 + 2)^2$ függvényt, a kezdőérték legyen $\mathbf{x}_0 = (1,1)^{\mathrm{T}}$. Használhatja például az

$$\mathbf{x}_{k+1} = \mathbf{x}_k - H_G(\mathbf{x}_k)^{-1} (\mathrm{grad}\, G(\mathbf{x}_k))^{\mathrm{T}}$$

iterációt (ekkor $\mathbf{x}_1 = (-\frac{17}{29}, \frac{17}{29})^{\mathrm{T}}$ lesz), vagy más minimalizációs eljárást.

22. fejezet

Numerikus Integrálás

22.1. Interpolációs kvadratúra-formulák

10.1.1. Feladat. Az I_n sorozat szigorúan monoton csökkenve 0-hoz tart, s megadható az $I_n = 1 - nI_{n-1}$, $I_0 = 1 - \frac{1}{e}$ rekurzív képlettel.

10.1.2. Feladat. *Útmutatás.* A b) esetben a rekurzió felerősíti, a c) esetben csökkenti a hibát.

10.1.3. Feladat. Az integrálokra teljesül az $I_n = 1 - 5I_{n-1}$, $I_0 = 1 - \ln\frac{6}{5}$ rekurzió.

10.1.4. Feladat. *Útmutatás.* Az integrálokra teljesül az $I_n + I_{n-1} = \frac{1}{n3^n}$ rekurzió.

10.1.5. Feladat. Az $\underline{r}_m$ Riemann-féle alsó, illetve a $\bar{r}_m$ Riemann-féle felső összegekre

$$\begin{aligned} \underline{r}_m &\le \int_a^b f(x)\,dx &\le \bar{r}_m \quad \text{és} \\ \underline{r}_m &\le \quad t_m &\le \bar{r}_m, \end{aligned}$$

ezért $\left|\int_a^b f(x)\,dx - t_m\right| \le \bar{r}_m - \underline{r}_m = \frac{b-a}{m}\left(f(b) - f(a)\right)$.

10.1.6. Feladat. A 10.1.5. Feladathoz hasonló egyenlőtlenségekkel oldható meg.

10.1.9. Feladat. *Útmutatás.*

$$\int_{-1}^{1} H_6(x)\,dx = \frac{7}{15}f(-1) + \frac{16}{15}f(0) + \frac{7}{15}f(1) + \frac{1}{15}\big(f'(-1) - f'(1)\big)$$

22.2. Gauss-kvadratúra

10.2.1. Feladat. Mivel ez az $n = 3$ alappontos Legendre-Gauss formula, pontos minden legfeljebb $2n-1$-ed fokú polinomra.

10.2.2. Feladat. Az integrál közelítő értéke $\int_0^1 \frac{\sin x}{x+1}\,dx \approx 0.2842269856$.

10.2.3. Feladat. Az integrál értéke $\int_0^\infty \frac{e^{-x}}{x+1}\,dx \approx 0.5963473622$. A kvadratúra formulával kapott közelítés $Q_3(f) \approx 0.5882352936$.

10.2.4. Feladat. Az integrál pontos értéke $\frac{1}{8}\frac{\pi^2-3}{\pi^2} \approx 0.08700455614$. A Gauss formulával kapott érték 0.08790654574.

10.2.5. Feladat. Az integrál értéke: $\int_{-1}^{1} x\left|\sin x\right| dx \approx 0.08700455614$. A Gauss formulával kapott érték $Q_3(f) \approx 0.2551118292$. Ezt és az előző feladatok eredményeit megkaphatja a programlisták a 155. oldalán található Maple munkalap segítségével is.
10.2.6. Feladat. Az elméletből ismert, hogy az $n=1,2,3,4,5,6,7,9$ esetekre létezik ilyen formula, de ha $n=8$ vagy $n>10$, akkor már nem.
A számításokhoz célszerű a Maple-t, vagy más programot segítségül hívni, a megfelelő Maple munkalap megtalálható a 156. oldalon.
10.2.8. Feladat. Lásd [17], 253. o.
10.2.9. Feladat. *Útmutatás.* A $q_1(x), q_2(x), \ldots, q_n(x)$ polinomok pontosan az $x_1, x_2, \ldots, x_n$ alappontokhoz tartozó Lagrange-féle bázispolinomok, mivel $q_j(x_k)=\delta_{jk}$. A fokszámok miatt $\langle q_j, g_k\rangle = Q_n(q_j(x)q_k(x))$ és $\langle xq_j, g_j\rangle = Q_n(xq_j(x)q_j(x))$.
Ennek fölhasználásával bizonyítható a három állítás.

22.3. Romberg integrálás

10.3.1. Feladat.

a) *Útmutatás.* Írja fel a deriváltakat tartalmazó hibatagot t_{2n}-re és t_n-re. Ha a deriváltak „kicsit változnak” csak, a két hibatagban közelítőleg ugyanaz a derivált-érték áll;

b) a)-hoz hasonlóan.

10.3.3. Feladat. A Romberg mátrix második és harmadik oszlopának fölírásakor pont ezekkel „korrigáljuk” az előző oszlop közelítéseit.
10.3.3. Feladat. Az integrál pontos értéke $I=\frac{1}{\sqrt{3}}$.
10.3.4. Feladat. Bár az integrálandó $f(x)=e^x\left|\sin(6\pi x)\right|$ függvény folytonos, elsőrendű deriváltja már nem az, sőt több szakadása is van $[a,b]$-ben. Az integrál kerekített értéke: $I \approx$ ≈ 256.066202, a Romberg-sémából $T_{8,8} \approx 255.8396080$, vagyis még ennek hibája is viszonylag nagy.
10.3.5. Feladat. A $k=1$ esetben a konvergencia valóban teljesül, hiszen az első oszlopban az összetett trapézösszegek állnak. Haladjunk tovább k szerinti indukcóval. Ha föltesszük, hogy a $1,2,\ldots,(k-1)$-dik oszlop elemeire igaz az állítás, akkor a $T_{j,k}=T_{j,k-1}+\frac{T_{j,k-1}-T_{j-1,k-1}}{4^{k-1}-1}$ rekurzív képletre a $j\to\infty$ határátmenetet alkalmazva kapjuk az állítást a k-dik oszlopra.
10.3.6. Feladat. A $t_{8,4}\approx 3{,}14159265$ közelítés minden jegye helyes.
10.3.8. Feladat. A Romberg algoritmust megvalósító függvény a programlisták 157. oldalán található.

22.4. Kvadratúra-sorozatok konvergenciája

10.4.1. Feladat. Legyen $a\le c<d\le b$ és tekintsük az

$$f(x)=\begin{cases}(x-c)(d-x) & \text{ha } x\in(c,d) \text{ és}\\ 0 & \text{különben}\end{cases}$$

folytonos függvényt. Ha föltennénk, hogy $[c, d]$ csak véges sok alappontot tartalmaz, akkor volna olyan m küszöbszám, hogy $n > m$ esetén már egyetlen Q_n formulának sem lenne alappontja $[c, d]$-ben, ekkor viszont csak $\lim_{n\to\infty} Q_n(f) = 0$ lehetne, holott $\int_a^b f(x)\,dx > 0$.
10.4.2. Feladat. Mivel Q_n rendje legalább 0, az $f(x) \equiv 1$ függvényre fölírható

$$\sum_{i=1}^{n} w_i^n = Q_n(1) = \int_a^b 1\rho(x)\,dx,$$

ahonnan a $K = \int_a^b \rho(x)\,dx$ konstanssal valóban következik a b) föltétel:

$$\sum_{i=1}^{n} \left|w_i^n\right| = \sum_{i=1}^{n} w_i^n = K.$$

10.4.4. Feladat. Mivel az n-dik Gauss formula rendje $2n-1$, a formulákkal kapott közelítések konvergensek minden rögzített $p(x)$ polinomra, tehát teljesül a Pólya-Steklov tétel a) föltétele. Másrészt minden formula pozitív, s ezért a 10.4.2. Feladat szerint teljesül a b) föltétel is.
10.4.7. Feladat. *Útmutatás.* A Riemann integrál definícióját felhasználva, az R_m osztáspontjaihoz tartozó tartozó Riemann-féle alsó- és felső összegekkel becsülheti R_m-et.
10.4.8. Feladat. Ezek a szabályok mind olyan pozitív Q alapformulára épülnek, amelynek rendje nagyobb 0-nál (nemcsak konstansokra pontos).

23. fejezet

Differenciálegyenletek megoldása

23.1. Taylor-sor és fokozatos közelítések módszere

11.1.1. Feladat. A pontos megoldást $Y(x)$-szel jelölve

a) A KÉP definíciója alapján $Y(0)=1$.

b) $Y'(x)=1-x+Y(x)$, tehát az $x=0$ helyen $Y'(0)=1-x_0+Y(x_0)=2$.

c) tovább differenciálva x szerint $Y''(x)=-1+Y'(x)=-1+(1-x+Y(x))$, vagyis $Y''(0)=1$,

Tehát a közelítő megoldás $Y(x)=1+2x+\frac{1}{2}x^2+\dots$ alakú lesz. A pontos megoldás $Y(x)=x+e^x$.
11.1.2. Feladat. *Útmutatás.* Igazolja, hogy $y^{(n)}(0)=(-1)^{n-1}\cdot 1\cdot 3\cdot 5\cdot\dots\cdot(2n-3)$, ha $n=2,3,\dots$
11.1.3. Feladat. Most $y_0(x)\equiv 1$, és

$$y_n(x)=1+\int_0^x t+(y_{n-1}(t))^2dt, \qquad n=1,2,\dots$$

Tehát

a) $y_1(x)=1+\int_0^x t+1dt=\left[\frac{t^2}{2}+t\right]_0^x=1+x+\frac{x^2}{2}$,

b) $y_2(x)=1+\int_0^x t+(1+t+\frac{t^2}{2})^2dt=\dots=1+x+\frac{3}{2}x^2+\frac{2}{3}x^3+\frac{1}{4}x^4+\frac{1}{20}x^5$,

c) $y_3(x)=1+\int_0^x t+(y_2(x))^2dt=\dots$

11.1.4. Feladat. A harmadik közelítés: $y_3(x)=\frac{2}{3}\sqrt{x^3}+\frac{x^4}{9}+\frac{x^9}{729}+\frac{8}{351}x^{\frac{13}{2}}$
11.1.5. Feladat. *Útmutatás.*

$$x(t)=\frac{e^{at}\sin(2t)a-2e^{at}\cos(2t)+2}{4+a^2},$$
$$y(t)=\frac{e^{at}\cos(2t)a-a+2e^{at}\sin(2t)}{4+a^2}.$$

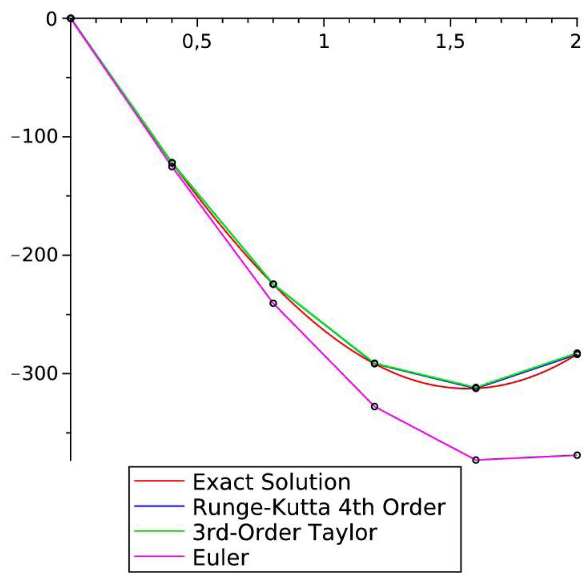
23.2. Runge-Kutta módszerek és lineáris többlépéses módszerek

11.2.1. Feladat. A pontos megoldás $Y(x) = \ln(x+1)$. A kapott két közelítés:

a) $y_3 \approx 0.27314$ és

b) $y_3 \approx 0.2719$.

A második módszer pontosabb eredményt ad, és még exponenciális függvénnyel sem kell számolni.
11.2.2. Feladat. Az egyetlen megoldás $Y(x) = \frac{x^3}{3}$. A közelítések nem konvergálnak.
11.2.3. Feladat. *Útmutatás.* Az $y_{n+1}(1+h\alpha b_{-1}) = \sum_{i=0}^{p}(a_i - h\alpha b_i)y_{n-i}$ alakban is fölírható a többlépéses módszerrel számított y_{n+1}, de a $d_i = a_i - h\alpha b_i$ jelölés bevezetése után ez ekvivalens a $\sum_{i=-1}^{p} d_i y_{n-i} = 0$ homogén differencia-egyenlet megoldásával.
11.2.4. Feladat. Az $\alpha = 0$, $\alpha = \frac{1}{2}$ és az $\alpha = 1$ esetekben az Euler-módszer speciális változatait kapjuk.
11.2.5. Feladat. Legyen például $y' = x^2 + y^2$, $y(0) = 0$.
11.2.6. Feladat. *Útmutatás.* A pontos megoldás $y(t) = t - 100\pi \sin t$. A közelítések viselkedését a 23.1. ábra mutatja.



23.1. ábra. A KÉP megoldásának numerikus közelítései

11.2.7. Feladat. *Útmutatás.* A pontos megoldás $Y(x) = \frac{1}{1+100x^2}$.
11.2.9. Feladat. *Útmutatás.* A módszer első karakterisztikus polinomja
$Q(z) = -z^2 + (1-\alpha)z + \alpha$. Ennek gyökei 1 és $-\alpha$. Tehát $-1 < \alpha \leq 1$-re teljesül a stabilitás.
11.2.10. Feladat. *Útmutatás.* Helyettesítsen be a rendet definiáló egyenletrendszer egyenleteibe. A rend tetszőleges α-ra legalább 3, és ha $\alpha = 1$, akkor 4.

23.3. Közönséges differenciálegyenletek peremérték-problémája

11.3.1. Feladat. Rajzolja fel az egyenlet $y(0) = 0$ és $y'(0) = 1$-et teljesítő megoldását, majd lássa be, hogy bármely α konstansra $z(x) = \alpha y(\alpha x)$ is megoldás.
11.3.2. Feladat. A pontos megoldás $y(x) = \sin x + \cos x$.
11.3.3. Feladat. Az intervallumot n részre osztva, a másodrendű deriváltakra az $y''(x_j) \approx \approx \frac{y(x_{j-1})-2y(x_j)+y(x_{j+1})}{h^2}$ közelítést alkalmazva az

$$\begin{aligned}
y_0 &= 0 \\
\frac{-y_0+2y_1-y_2}{h^2}+\frac{1}{2}(y_1+h+1)^3 &= 0 \\
&\vdots \\
\frac{-y_{j-1}+2y_j-y_{j+1}}{h^2}+\frac{1}{2}(y_j+jh+1)^3 &= 0 \qquad j=1,\ldots,n-1 \\
y_n &= 0
\end{aligned}$$

egyenletrendszert kapjuk. $2h^2$-tel beszorozva és rendezve a kapott egyenletrendszert a többváltozós Newton-módszerrel közelíthetjük az $(x_0, x_1, x_2, \ldots, x_n)^T$ pontokban a rendszer $(y_0, y_1, y_2, \ldots, y_n)^T$ megoldását.
A pontos megoldás az $y(x) = \frac{x-x^2}{x-2}$ függvény.

24. fejezet

Numerikus programkönyvtárak használata

24.1. A GSL és a LAPACK

12.1.1. Feladat. A megoldást kiszámoló, a programlisták 158. oldalán található program az $\mathbf{x}^* = (-1,\ -2,\ -2,\ 0)^{\mathrm{T}}$ pontos megoldást írja ki.
12.1.2. Feladat. A felbontást kiszámoló program a programlisták 159. oldalán található.
12.1.3. Feladat. A sajátérték-feladatot megoldó program a programlisták 160. oldalán található.
12.1.4. Feladat. A spline-közelítést kiszámoló program a programlisták 161. oldalán található.
12.1.5. Feladat. A megoldást kiszámoló program a programlisták 162. oldalán található.
12.1.6. Feladat.

a) a pontos érték: -4, a kapott közelítés: -3.999999999999996891;

b) az integrál értéke ≈ 0.9078007611, a kapott közelítés: $0.9077954465602833 26$;

c) a pontos érték: π, a közelítés: 3.141592653589301065.

A megoldást kiszámoló C program a programlisták 163. oldalán található.
12.1.7. Feladat. A megoldást kiszámoló program a programlisták 164. oldalán található. A kapott $\mathbf{s} = (17.423768,\ 6.653077,\ 3.483494,\ 0.118867)^{\mathrm{T}}$ értékek jól közelítik a Maple-lel elért eredményt.
12.1.8. Feladat. Az a) rész megoldása a programlisták 165. oldalán található.

Harmadik rész

Programlisták

A 2.1.20. **Feladat** programja:

```
function Gauss(A,b)
  sA=size(A);
  sb=size(b);
  if sA(1)~=sA(2) | sA(1)~=sb(1) | sb(2)~=1
    printf('Nem jók a paraméterek!')
    abort
  end
  n=sA(1);
  A=[A,b];
  disp('0. lépés:')
  disp(A)
  for i=1:n-1
    if A(i,i)== 0
      printf('A főátlóban 0 érték van, az elimináció leáll!')
      abort
    end
    for j=i+1:n
      kiv=A(j,i)/A(i,i);
      A(j,:)=A(j,:)-kiv*A(i,:);
    end
    printf('%d. lépésben az iterációs mátrix:\n',i)
    disp(A)
  end
  if A(n,n)==0 &  A(n,n+1)~=0
     printf('Nincs megoldás!')
     abort
     elseif A(n,n)==0 &  A(n,n+1)==0
     x(n)=1
  else
  x(n)=A(n,n+1)/A(n,n);
  for i=n-1:-1:1,
    x(i)=A(i,n+1);
    for j=i+1:n,
        x(i)=x(i)-A(i,j)*x(j);
    end
    x(i)=x(i)/A(i,i);
  end
disp('A megoldás:')
disp(x)
endfunction
```

A 2.1.21. **Feladat** programja:

```
function GaussJordan(A,b)
  sA=size(A);
  sb=size(b);
  if sA(1)~=sA(2) | sA(1)~=sb(1) | sb(2)~=1
    printf('Nem jók a paraméterek!')
    abort
  end
  n=sA(1);
  A=[A,b];
  disp('0. lépés:')
  disp(A)
  for i=1:n
    if A(i,i)== 0
      printf('Főátlóban 0 érték van, az elimináció leáll!')
      abort
    end
    for j=1:n
      if i~=j
        kiv=A(j,i)/A(i,i);
        A(j,:)=A(j,:)-kiv*A(i,:);
      end
    end
    printf('%d. lépésben az iterációs mátrix:\n',i)
    disp(A)
  end
  for i=1:n
    A(i,:)=A(i,:)/A(i,i)
  end
  printf('A megoldás az alábbi mátrixból olvasható ki:\n')
  disp(A)
endfunction
```

A 2.2.19. **Feladat** programja:

```
function GaussSeidel(A,b,x,t)
sA=size(A);
sb=size(b);
if sA(1) ~= sA(2) | sA(1) ~= sb(1) | sb(2) ~= 1
    disp('Rossz dimenziók!')
    abort
end
n=sA(1);
printf('0. lépés:');
disp(x)
l=1;
p=x+2*t;
while abs(x-p)>t & l<100,
   p=x;
   for i=1:n
      x(i)=b(i);
      for j=1:i-1
          x(i)=x(i)-A(i,j)*x(j);
      end
      for j=i+1:n
          x(i)=x(i)-A(i,j)*p(j);
      end
      x(i)=x(i)/A(i,i);
   end
   printf('%d. lépés:',l);
   disp(x)
   l=l+1;
end
endfunction
```

A 3.1.14. **Feladat** programja:

```
function LagrIntV(x,y)
if length(x)~=length(y)
  printf('Rossz vektorokat adott meg!')
  abort
end
n=length(x);
for i=1:n
   for j=1:n
      V(i,j)=x(i)^(j-1);
   end
end
if det(V)==0
   disp('Két azonos x értéket adott meg!')
   abort
else
   disp('A Vandermonde mátrix:')
   disp(V)
   disp('A Vandermonde mátrix determinánsa:')
   disp(det(V))
   a=V\y;
   disp('A polinom együtthatói:')
   disp(a)
end
endfunction
```

A 3.1.23. **Feladat** programja:

```
function NewtIntp(x,y)
if length(x)~=length(y)
  printf('Rossz vektorokat adott meg!')
  abort
end
a=y;
n=length(x);
for k = 1 : n - 1
   d(k, 1) = (y(k+1) - y(k))/(x(k+1) - x(k));
end
for j = 2 : n - 1
   for k = 1 : n - j
      d(k, j) = (d(k+1, j - 1) - d(k, j - 1))/(x(k+j) - x(k));
   end
 end
disp('Az osztott differenciák táblázata:')
disp(d)
for j=2:n
    for i=n:-1:j
        a(i)=(a(i)-a(i-1))/(x(i)-x(i-j+1));
    end
end
disp('A Newton-polinom együtthatói: ')
disp(a)
endfunction
```

A 3.2.5. **Feladat** programja:

```
function Hermint(x,y,yd)
n=length(x);
foszt=zeros(2*n);
for i=1:n
    xx(2*i-1)=x(i);
    xx(2*i)=x(i);
foszt(2*i-1,1)=y(i);
foszt(2*i,1)=y(i);
foszt(2*i,2)=yd(i);
end
    for i=2:n
       foszt(2*i-1,2)=(foszt(2*i-1,1)-foszt(2*i-2,1))/(xx(2*i-1)-xx(2*i-2));
    end
for j=3:2*n
    for i=j:2*n
       foszt(i,j)=(foszt(i,j-1)-foszt(i-1,j-1))/(xx(i)-xx(i-j+1));
    end
end
printf('A megoldás:'')
disp(foszt)
endfunction
```

A 4.1.7. **Feladat** programja:

```
function Intfel(f,a,b,t)
if f(a)*f(b)>0
   printf('Nem ellentétes előjelű az intervallum!')
   abort
end
if f(a)==0
   printf('Az intervallum első végpontja megoldás!')
   abort
end
if f(b)==0
   printf('Az intervallum második végpontja megoldás!')
   abort
end
l=1;
c=(a+b)/2;
while abs(f(c))>=t & l < 100
   c=(a+b)/2;
   if f(a)*f(c) < 0
      b=c;
   elseif f(b)*f(c) < 0
      a=c;
   else
      printf('%d. lépésben a pontos megoldás: %1.14f\n',l,c)
      abort
   end
   printf('%d. lépésben a közelítő megoldás: %1.14f\n',l,c)
   l=l+1;
 end
endfunction
```

A 4.1.8. **Feladat** programja:

```
function Hur(f,a,b,t)
if f(a)*f(b)>0
   printf('Nem ellentétes előjelű az intervallum!')
   abort
end
if f(a)==0
   printf('Az intervallum első végpontja megoldás!')
   abort
end
if f(b)==0
   printf('Az intervallum második végpontja megoldás!')
   abort
end
l=1;
c=a-f(a)*(a-b)/(f(a)-f(b));
while abs(f(c))>=t & l < 100
   c=a-f(a)*(a-b)/(f(a)-f(b));
   if f(a)*f(c) < 0
      b=c;
   elseif f(b)*f(c) < 0
      a=c;
   else
      printf('%d. lépésben a pontos megoldás: %1.14f\n',l,c)
      abort
   end
   printf('%d. lépésben a közelítő megoldás: %1.14f\n',l,c)
   l=l+1;
end
endfunction
```

A 4.2.8. **Feladat** programja:

```
function Newton(f,df,a,t)
if f(a)==0 | abs(f(a))<t
   printf('A kezdőpont megoldás!')
   abort
end
l=1;
while abs(f(a))>=t & l < 100
   if df(a) == 0
      printf('Nem található megoldás, az érintő párhuzamos az $x$ tengellyel!')
      abort
   end
   a=a-f(a)/df(a);
   printf('%d. lépés után a közelítő megoldás: %1.14f\n',l,a)
   l=l+1;
end
endfunction
```

A 4.2.9. **Feladat** programja:

```
function Szelo(f,a,b,t)
if f(a)==0
   printf('Az első kezdőpont megoldás!')
   abort
end
if f(b)==0
   printf('A második kezdőpont megoldás!')
   abort
end
l=1;
if f(a)-f(b) == 0
  printf('Nem található megoldás, a szelő párhuzamos az $x$ tengellyel!')
  abort
end
c=b-f(b)*(b-a)/(f(b)-f(a));
while abs(f(c))>=t & l < 100
   if f(a)-f(b) == 0
     printf('Nem található megoldás, a szelő párhuzamos az $x$ tengellyel!')
     abort
   end
   c=b-f(b)*(b-a)/(f(b)-f(a));
   printf('%d. lépésben a közelítő megoldás: %1.14f\n',l,c)
   a=b;
   b=c;
   l=l+1;
end
endfunction
```

A 4.3.7. **Feladat** programja:

```
function Fixpont(f,a,t)
l=1;
while abs(a-f(a)) >= t & l < 100
  a=f(a)
  printf('%d lépésben a közelítő megoldás: %1.14f\n',l,a)
  l=l+1;
end
endfunction
```

Az 5.2.9. **Feladat** programja:

```
function Trapez(f,a,b,n)
h=(b-a)/n;
for i=1:n+1
   x(i)=a+h*(i-1);
end
mo=f(a)+f(b);
for i=2:n
   mo=mo+2*f(x(i));
end
mo=h/2*mo
printf('A közelítő megoldás %d részre osztva az intervallumot: %1.14f\n',n,mo)
endfunction
```

Az 5.2.14. Feladat programja:

```
function Simpson(f,a,b,n)
if pmodulo(n,2)~=0
   printf('A negyedik paraméternek párosnak kell lennie!')
   abort
end
h=(b-a)/n;
for i=1:n+1
   x(i)=a+h*(i-1);
end
mo=f(a)+f(b);
mo1=0;
for i=1:n/2
   mo1=mo1+f(x(2*i));
end
mo2=0;
for i=1:(n/2-1)
   mo2=mo2+f(x(2*i+1));
end
mo=h/3*(mo+2*mo1+4*mo2)
printf('A közelítő megoldás %d részre osztva az intervallumot: %1.14f\n',n,mo)
endfunction
```

A 6.3.7. **Feladat** programja:

```
function OptGrad(x,A,b,t)
g=-A*x+b;
i=0;
while norm(g,2)>=t
    lambda=g'*g/((A*g)'*g);
    gr=g;
    xr=x;
    x=xr+lambda*g
    i=i+1;
    g=-A*x+b;
end
printf('%d lépés után a közelítő megoldás: ',i)
disp(g)
endfunction
```

A 6.3.11. **Feladat** programja:

```
function KonGrad(x,A,b,t)
g=A*x-b;
d=g;
i=0;
while norm(g,2)>=t
    lambda=-g'*d/((A*d)'*d);
    gr=g;
    xr=x;
    x=xr+lambda*d
    i=i+1;
    g=gr+lambda*A*d;
    mu=((A*g)'*g)/((A*d)'*d);
    d=-g+mu*d;
end
printf('%d lépés után a közelítő megoldás: ',i)
disp(g)
endfunction
```

A 7.1.18. **Feladat** programja:

```
QRTukr := proc( A::'Matrix'(shape=square, float) )
local n, Q, R, j, a_j, h_j, Hhat_j;
uses LinearAlgebra;
Digits := myDigits;
  n := RowDimension(A);
  R := Copy(A);
  Q := Matrix( IdentityMatrix(n) );
  for j from 1 to n-1 do
    a_j := Vector(R[j..n,j]);
    h_j := a_j - sign(a_j[1])*VectorNorm(a_j,2)*UnitVector(1,n+1-j);
    Hhat_j := HouseholderMatrix(h_j);
    R[j..n,j..n] := Hhat_j.R[j..n,j..n];
    Q[j..n,j..n] := Q[j..n,j..n].Hhat_j;
    if j > 1 then
      Q[1..j-1,j..n] := Q[1..j-1,j..n].Hhat_j;
    end if;
  end do;
return Q, R;
end proc:
##   Tesztelés véletlenmátrixokkal:
#with(LinearAlgebra):
#interface(displayprecision=3);
#n        := 5;
#myDigits := 20;
#A        := RandomMatrix(n,outputoptions=[datatype=float]);
#Q,R      := Householder_QR(A);
#   ellenőrzés:
#A - Q.R;
#   ugyanez a Maple eljárásával:
#Q_M, R_M := QRTukr(A);
#A-Q_M.R_M;
#A:= RandomMatrix(n, n+1,outputoptions=[datatype=float]);
#Q,R      := QRTukr(A);
#Q_M, R_M := QRDecomposition(A);
#A-Q_M.R_M;
```

A 7.1.20. Feladat programja:

```
QRForg := proc(A::'Matrix'(shape=square, float))
local myeps, n, Q, R, d, cos_theta, sin_theta, a_ik, a_jk, q_ki, q_kj, i, j, k;
uses LinearAlgebra;
myeps := 1/2* 10^(-myDigits);
Digits := myDigits;
n := RowDimension(A);
R := Copy(A);
Q := Matrix(IdentityMatrix(n));
for i from 1 to n do
  for j from i+1 to n do
    d := sqrt(R[i,i]^2 + R[j,i]^2);
    if abs(d) < myeps then
      break
    end if;
    cos_theta := R[i,i]/d;
    sin_theta := R[j,i]/d;
    for k to n do
      a_ik := R[i,k];
      a_jk := R[j,k];
      R[i,k] :=   cos_theta * a_ik + sin_theta * a_jk;
      R[j,k] := - sin_theta * a_ik + cos_theta * a_jk;
      q_ki := Q[k,i];
      q_kj := Q[k,j];
      Q[k,i] :=   cos_theta * q_ki + sin_theta * q_kj;
      Q[k,j] := - sin_theta * q_ki + cos_theta * q_kj;
    end do;
  end do;
end do;
return Q, R:
end proc:
##   Tesztelés véletlen mátrixokkal:
#with(LinearAlgebra):
#interface(displayprecision=4);
#n        := 5;
#myDigits := 5;
#A        := RandomMatrix(n,outputoptions=[datatype=float]);
#Q,R      := QRForg(A);
##   ellenőrzés:
#A - Q.R;
#   ugyanez a Maple eljárásával:
#Q_M, R_M := QRDecomposition(A);
#A-Q_M.R_M;
#n        := 5;
#myDigits := 20;
#A        := RandomMatrix(n,outputoptions=[datatype=float]);
#Q,R      := QRForg(A);
##   ellenőrzés:
#A - Q.R;
##   ugyanez a Maple eljárásával:
#Q_M, R_M := QRDecomposition(A);
#A-Q_M.R_M;
##  Hilbert mátrixokkal:
#Hfl := evalf(HilbertMatrix(5));
#for j in [4,8,12,16] do
#  myDigits := j;
#  Q, R := QRForg(Hfl):
#  Hfl  - Q.R;
#end do;
```

A 7.4.9. **Feladat** programja:

```
LRTransz := proc(A::'Matrix'(float,square),kMAX::posint,eps::positive)
local A1, P1, L1, R1, MaxElem, i, k, l, n;
uses LinearAlgebra;
Digits := myDigits;
A1 := Copy(A);
n := RowDimension(A);
for k from 1 while k < kMAX do
    MaxElem := 0;
    for i from 2 to n do
      for l from 1 to i-1 do
        if (abs(A1[i,l]) > MaxElem) then
        MaxElem := abs(A1[i,l]);
        end if;
      end do;
    end do;
    if MaxElem > eps then
        P1, L1, R1 :=LUDecomposition( A1,method='GaussianElimination' );
        A1 := R1.P1.L1;
    else
          return k, MaxElem, Vector( n,[seq(A1[i,i],i=1..n)] );
    end if;
end do;
return k, MaxElem, "Nem konvergált az algoritmus";
end proc:

## Teszteljük az eljárást véletlen mátrixokkal:
#with(LinearAlgebra):
#myDigits := 25;
#n := 4;
#Lambda := RandomMatrix(n,generator=-10..10, outputoptions=[datatype=float, shape=diagonal]):
#T :=  RandomMatrix(n,generator=-10..10, outputoptions=[datatype=float]):
#A := MatrixInverse(T).Lambda.T;
#Eigs := sort(Eigenvalues(A));
#k, Maxelem, Diag := LRTransz(A, 10, 0.001);
#k, Maxelem, Diag := LRTransz(A, 1000, 0.00001);
#printf(" A lépésszám: k = %a \n",k);
#printf(" Az A_k mátrix főátló alatti maximális eleme: %a \n", Maxelem);
#printf(" A sajátértékek   közelítései:\n");
#print(sort(Diag));
#B := evalf(RandomMatrix(n, outputoptions=[float, shape=symmetric]));
#Eigs := sort(Eigenvalues(B));
#k, Maxelem, Diag := LRTransz(B, 10, 0.001);
#k, Maxelem, Diag := LRTransz(B, 1000, 0.00001);
#printf(" A lépésszám: k = %a \n",k);
#printf(" A B_k  mátrix főátló alatti maximális eleme: %a \n", Maxelem);
#printf(" A sajátértékek   közelítései:\n");
#print(sort(Diag));
## Erre a C mátrixra nem konvergálhat az LR transzformácó:
#C := Matrix([[5., -4., -9., 7.], [-3., 6., 4., -9.], [10., 7., 3., -8.], [3., 4., 2., 8.]]);
#Eigs := sort(Eigenvalues(C));
#k, Maxelem, Diag := LRTransz(C, 1000, 0.001);
```

A 7.4.11. **Feladat** programja:

```
QRTranszS := proc(A::'Matrix'(float,square),kMAX::posint,eps::positive)
local A1, Q1, R1, MaxElem, i, k, l, n, sigma;
uses LinearAlgebra;
A1 := Copy(A);
n := RowDimension(A);
for k from 1 while k < kMAX do
    MaxElem := 0;
    for i from 2 to n do
      for l from 1 to i-1 do
        if (abs(A1[i,l]) > MaxElem) then
        MaxElem := abs(A1[i,l]);
        end if;
      end do;
    end do;
    if MaxElem > eps then
          sigma := A1[n,n];
        Q1, R1 := QRDecomposition( A1-sigma*IdentityMatrix(n),fullspan );
        A1 := R1.Q1 + A1[n,n]*IdentityMatrix(n);
    else
          return k, MaxElem, Vector( n,[seq(A1[i,i],i=1..n)] );
    end if;
end do;
return k, MaxElem, "Nem konvergált az algoritmus";
end proc:
## Teszteljük az eljárást:
#with(LinearAlgebra):
#n := 4;
#Lambda := RandomMatrix(n,generator=-10..10, outputoptions=[datatype=float, shape=diagonal]):
#T :=  RandomMatrix(n,generator=-10..10, outputoptions=[datatype=float]):
#A := MatrixInverse(T).Lambda.T;
#Eigs := sort(Eigenvalues(A));
#k, Maxelem, Diag := QRTranszS(A, 10, 0.001);
#k, Maxelem, Diag := QRTranszS(A, 1000, 0.00001);
#printf(" A lépésszám: k = %a \n",k);
#printf(" Az A_k mátrix főátló alatti maximális eleme: %a \n", Maxelem);
#printf(" A sajátértékek   közelítései:\n");
#print(sort(Diag));
#B := evalf(RandomMatrix(n, outputoptions=[float, shape=symmetric]));
#Eigs := sort(Eigenvalues(B));
#k, Maxelem, Diag := QRTranszS(B, 10, 0.001);
#k, Maxelem, Diag := QRTranszS(B, 1000, 0.00001);
#printf(" A lépésszám: k = %a \n",k);
#printf(" A B_k mátrix főátló alatti maximális eleme: %a \n", Maxelem);
#printf(" A sajátértékek   közelítései:\n");
#print(sort(Diag));
## Erre a C mátrixra nem konvergálhat a QR transzformáció:
#C := Matrix([[5., -4., -9., 7.], [-3., 6., 4., -9.], [10., 7., 3., -8.], [3., 4., 2., 8.]]);
#Eigs := sort(Eigenvalues(C));
#k, Maxelem, Diag := QRTranszS(C, 1000, 0.001);
```

A 8.1.8. Feladat programja:

```
HaarMatrix := proc(G, X)
local n, i, j, Hmx;
uses LinearAlgebra;
  n := Dimension(G);
  Hmx := Matrix(n, (i,j) -> G[j](X[i]));
  return Hmx;
end proc:
LBazis := proc(G, X)
local n, i,j, k, Ljmx, LB;
uses LinearAlgebra;
  n := Dimension(G);
  for j from 1 to n do
    Ljmx := Matrix(n, (i,k) -> if not i=j then G[k](X[i]) else G[k] end if);
    LB[j] := Determinant(Ljmx)/Determinant(HaarMatrix(G, X));
  end do;
return LB;
end proc:
AltInt := proc( G, f, X)
local n,F,L;
uses LinearAlgebra;
n := Dimension(G);
F := Vector( [seq(evalf(f(X[i])), i=1..n)] );
  L := LBazis(G, X);
  return add( F[i]*evalf(L[i]), i = 1..n );
end:
# Tesztelés különböző példákkal:
#n := 4;
#G := Vector[row]([x->1,x ->x,x->x^2,x->x^3]);
#X := evalf(Vector[row]([ seq(i, i=1..n) ]));
#f:= x -> sqrt(x+1);
#F := evalf(Vector[row]([seq(f(X[i]), i=1..n)]) );
#P4 := AltInt( G, f, X);
#P4(1); P4(1.5);
#plot([f(x) -P4(x)], x= 0..5, legend="f(x) - P4(x)",title="Az általánosított interpoláció hibája\n");
#n := 5;
#G := Vector[row]([x->1,x ->sin(x),x->cos(x),x->sin(2*x),x ->cos(2*x)]);
#X := evalf(Vector[row]([ seq(i*2*Pi/n, i=1..n) ]));
#f:= x -> x*sin(x);
#F := evalf(Vector[row]([seq(f(X[i]), i=1..n)]) );
#P5 := AltInt( G, f, X);
#evalf(P5(1)); P5(Pi);
#plot([f(x) -P5(x)], x= 0..2*Pi, legend="f(x) - P5(x)",title="Az általánosított interpoláció hibája\n");
#n := 5;
#G := Vector[row]([x->1,x ->exp(x),x->exp(2*x),x->exp(3*x),x ->exp(4*x)]);
#X := evalf(Vector[row]([ seq(i, i=0..n-1) ]));
#f:= x -> exp(x+1); # ez éppen az exp(1)*exp(x) függvény...
#F := evalf(Vector[row]([seq(f(X[i]), i=1..n)]) );
#P5 := AltInt( G, f, X);
#P5(0);evalf(P5(1));
#plot([f(x) -P5(x)], x= 0..6, legend="f(x) - P5(x)",title="Az általánosított interpoláció hibája\n");
```

A 8.4.6. Feladat programja:

```
OrtoPoliG := proc(a, b, rho, n)
local b_szorzat, p_list, a_list, i, j;
b_szorzat := proc(f, h)
return int(f*h*rho, x = a..b);
end proc:
p_list := [ 1 ];
for i from 1 to n do
a_list := [ seq( (-1)* b_szorzat(x^i, p_list[j])/b_szorzat(p_list[j], p_list[j]), j = 1..i )];
p_list := [ op(p_list),  x^i + add( a_list[j]*p_list[j], j = 1..i ) ];
end do;
return p_list[n+1];
end proc:
# Tesztelés különböző paraméterekkel:
#a := -1: b := 1:        rho := 1:             n := 4:  P_lista := [seq (OrtoPoliG(a,b,rho,j), j=1..n)];
#a := -1: b := 1:        rho := 1/sqrt(1-x^2): n := 4:  P_lista := [seq (OrtoPoliG(a,b,rho,j), j=1..n)];
#a :=  0: b := infinity: rho := exp(-x):       n := 4:  P_lista := [seq (OrtoPoliG(a,b,rho,j), j=1..n)];
```

A 8.4.7. **Feladat** programja:

```
OrtoPoliL := proc(a, b, rho, n)
local b_szorzat, p_list, i, beta, gamma ;
b_szorzat := proc(f, h)
return int(f*h*rho, x = a..b);
end proc:
p_list := [ 1 ];
for i from 0 to n do
  beta := b_szorzat(x*p_list[i+1], p_list[i+1])/b_szorzat(p_list[i+1], p_list[i+1]);
  if i = 0 then
    gamma := 0
  else
    gamma := b_szorzat(p_list[i+1], p_list[i+1])/b_szorzat(p_list[i], p_list[i])
  end if;
  if i = 0 then
    p_list := [ op(p_list), (x - beta)*p_list[i+1] ];
  else
    p_list := [ op(p_list), (x - beta)*p_list[i+1] - gamma *p_list[i] ];
  end if;
end do;
return simplify(p_list[n+1]);
end proc:
# Tesztelés különböző paraméterekkel:
#a := -1: b := 1:        rho := 1:             n := 4:  P_lista := [seq (OrtoPoliL(a,b,rho,j), j=1..n)];
#a := -1: b := 1:        rho := 1/sqrt(1-x^2): n := 4:  P_lista := [seq (OrtoPoliL(a,b,rho,j), j=1..n)];
#a :=  0: b := infinity: rho := exp(-x):       n := 4:  P_lista := [seq (OrtoPoliL(a,b,rho,j), j=1..n)];
```

A 9.1.16. **Feladat** programja:

```
LinJOR := proc(A::'Matrix'(float), b::'Vector'(float), x0::'Vector'(float),
               w::positive, kMAX::posint, eps::positive)
  local i, j,k, n, xold, xnew;
  uses LinearAlgebra;
  Digits := myDigits;
  n :=  RowDimension( A );
  xold := copy(x0);
  xnew := copy(x0);
  for j from 1 to kMAX  do
    for i from 1 to n do
      xnew[i]:= -(w/A[i,i]) * (add(A[i,k]*xold[k], k=1..i-1) + add(A[i,k]*xold[k], k=i+1..n) - b[i])
                 + (1-w)*xold[i];
    end do;
    if VectorNorm(b - A.xold) >= eps then
      xold := copy(xnew);
    else
      return j, xnew;
    end if;
  end do;
  return j, xnew, "Nem konvergált az iteráció";
end proc:
# Teszteljük az eljárást:
#with(LinearAlgebra):
#A := Matrix([[3.,2.,1.],[1.,4.,1.],[1.,1.,5.]]);
#b := Vector([6.,6.,7.]);
#x_pontos := LinearSolve( A, b );
#myDigits := 10;
#x0 := Vector(3,[ 0.5, 0.5, 0.]);
#LinJOR( A, b, x0, 0.05, 100, 0.0001);
#LinJOR( A, b, x0, 0.05, 500, 0.0001);
#LinJOR( A, b, x0, 0.85, 100, 0.0001);
#LinJOR( A, b, x0, 1.85, 100, 0.0001);
```

A 9.1.18. **Feladat** programja:

```
LinSOR := proc(A::'Matrix'(float), b::'Vector'(float), x0::'Vector'(float),
               w::positive, kMAX::posint, eps::positive)
  local i, j, k, n, xold, xnew;
  uses LinearAlgebra;
  Digits := myDigits;
  n :=  RowDimension( A );
  xold := copy(x0);
  xnew := copy(x0);
  for j from 1 to kMAX  do
    for i from 1 to n do
      xnew[i]:= -(w/A[i,i])*(add(A[i,k]*xnew[k], k=1..i-1) + add(A[i,k]*xold[k], k=i+1..n) - b[i])
                 + (1-w)*xold[i];
    end do;
    if VectorNorm(b - A.xold) >= eps then
      xold := copy(xnew);
    else
      return j, xnew;
    end if;
  end do;
  return j, xnew, "Nem konvergált az iteráció";
end proc:
## Teszteljük az eljárást:
#with(LinearAlgebra):
#A := Matrix([[3.,2.,1.],[1.,4.,1.],[1.,1.,5.]]);
#b := Vector([6.,6.,7.]);
#x_pontos := LinearSolve( A, b );
#myDigits := 10;
#x0 := Vector(3,[ 0.5, 0.5, 0.]);
#LinSOR( A, b, x0, 0.05, 100, 0.0001);
#LinSOR( A, b, x0, 0.05, 500, 0.0001);
#LinSOR( A, b, x0, 0.85, 100, 0.0001);
#LinSOR( A, b, x0, 1.85, 100, 0.0001);
```

A 9.2.12. **Feladat** programja:

```
Fixpont := proc(G::Vector, x0::'Vector'(float), kMAX::posint, eps)
local n, k, x_k, x_k1, i;
uses LinearAlgebra;
Digits := myDigits;
n := Dimension(x0):
x_k := Copy(x0):
x_k1 := Vector(n):
for k from 1 to kMAX do
  for i from 1 to n  do
     x_k1[i] := evalf( subs( seq(x[j]=x_k[j], j=1..n), G[i]) ):
  end do:
if VectorNorm(x_k1-x_k, infinity) < eps then
return k, x_k1:
else
 x_k := Copy(x_k1):
end if:
end do:
return k-1, x_k1, "Nem konvergált az iteráció";
end proc:
## Teszteljük az eljárást:
#with(LinearAlgebra):
#G := Vector([sin(x[1])*sin(x[2])*sin(x[3])-1/2,1/4*sqrt(x[1]^2+x[2]^2+x[3]^2),1/2*(x[1]^2+x[2]^2)] );
#x0 := Vector( [2., 2., 2.] );
#myDigits:=20;
#k, x_k :=Fixpont(G, x0, 25, 0.00000001):
#printf(" Az iterációs lépések száma: k= %a\n", k);
#printf(" A megoldásra kapott x_k közelítés:\n");
#print(x_k);
## Ellenőrzés a Maple fsolve eljárásával:
#fsolve({x[1] = G[1], x[2] = G[2], x[3] = G[3]}, {x[1],x[2],x[3]} );
```

A 9.3.10. **Feladat** programja:

```
NewtonSys := proc(JacobiMX::Matrix(shape=square), F::Vector, x0::'Vector'(float),
                  kMAX::posint, eps1::positive, eps2::positive)
local n, k,x_k, x_k1, Jacobi_k, F_k, s_k;
uses LinearAlgebra;
Digits := myDigits;
n := RowDimension(JacobiMX);
x_k := Copy(x0);
for k from 1 to kMAX do
    Jacobi_k:= subs({seq(x[j]=x_k[j], j=1..n)},JacobiMX);
    F_k:= subs({seq(x[j]=x_k[j], j=1..n)},F);
    s_k:= LinearSolve(Jacobi_k,-F_k);
    x_k1 := x_k + s_k;
    if (VectorNorm(F_k) < eps1) and (VectorNorm(x_k1 - x_k) < eps2) then
        return x_k1;
    else
        x_k := x_k1;
    end if;
end do;
return  k, x_k1, "Nem konvergált az iteráció";
end proc:
## Teszteljük az eljárást:
#n := 2;
#myDigits := 15;
#F:=Vector([x[1]^2+x[2]^2-1, x[1]*x[2] - 2*x[1]^2+1]);
#J:=VectorCalculus[Jacobian](F,[x[1],x[2]]):
#solve({F[1], F[2]}):
#evalf(allvalues(%));
#x0:=Vector([-0.84, -0.5]);
#NewtonSys(J, F, x0, 10, 0.0000001, 0.0000001);
#F:=Vector([x[1]^2+x[2]^2-1, x[2] - 2*x[1]]):
#J:=VectorCalculus[Jacobian](F,[x[1],x[2]]):
#solve({F[1], F[2]}):
#evalf(allvalues(%));
#x0:=Vector([0.5, 1.0]);
#NewtonSys(J, F, x0, 20, 0.001, 0.001);
#x0:=Vector([-0.5, -1.0]);
#NewtonSys(J, F, x0, 20, 0.001, 0.001);
#x0:=Vector([1.0, -1.0]);
#NewtonSys(J, F, x0, 20, 0.001, 0.001);
```

A 9.3.11. **Feladat** programja:

```
NewtonSysS := proc(Jacobi_0::Matrix(shape=square), F::Vector, x0::'Vector'(float),
                   kMAX::posint, eps1::positive, eps2::positive)
local n, k, x_k, x_k1, F_k, s_k;
uses LinearAlgebra;
Digits := myDigits;
n := RowDimension(Jacobi_0);
x_k := evalf(x0);
for k from 1 to kMAX do
    F_k:= subs({seq(x[j]=x_k[j], j=1..n)},F);
    s_k:= LinearSolve(Jacobi_0, -F_k);
    x_k1 := x_k + s_k;
    if (VectorNorm(F_k) < eps1) and (VectorNorm(x_k1 - x_k) < eps2) then
        return x_k1;
    else
        x_k := x_k1;
    end if;
end do;
return k, x_k1, "Nem konvergált az iteráció";
end proc:
## Teszteljük az eljárást:
#n := 2;
#myDigits := 15;
#F:=Vector([x[1]^2+x[2]^2-1, x[1]*x[2] - 2*x[1]^2+1]);
#J:=VectorCalculus[Jacobian](F,[x[1],x[2]]);
#solve({F[1], F[2]});
#evalf(allvalues(%));
#x0:=Vector([-0.84, -0.5]);
#J_0 := subs({seq(x[j]=x0[j], j=1..n)}, J);
#NewtonSysS(J_0, F, x0, 10, 0.001, 0.001);
#x0:=Vector([-0.84, -0.5]);
#J_0 := subs({seq(x[j]=x0[j], j=1..n)}, J);
#NewtonSysS(J_0, F, x0, 10, 0.0000001, 0.0000001);
#F:=Vector([x[1]^2+x[2]^2-1, x[2] - 2*x[1]]);
#J:=VectorCalculus[Jacobian](F,[x[1],x[2]]);
#solve({F[1], F[2]});
#evalf(allvalues(%));
#x0:=Vector([0.5, 1.0]);
#J_0 := subs({seq(x[j]=x0[j], j=1..n)}, J);
#NewtonSysS(J_0, F, x0, 20, 0.001, 0.001);
#x0:=Vector([-0.5, -1.0]);
#J_0 := subs({seq(x[j]=x0[j], j=1..n)}, J);
#NewtonSysS(J_0, F, x0, 20, 0.001, 0.001);
#x0:=Vector([0.5, 0.25]);
#J_0 := subs({seq(x[j]=x0[j], j=1..n)}, J);
#NewtonSysS(J_0, F, x0, 20, 0.001, 0.001);
#x0:=Vector([1.0, -1.0]);
#J_0 := subs({seq(x[j]=x0[j], j=1..n)}, J);
#NewtonSysS(J_0, F, x0, 20, 0.001, 0.001);
```

A 9.3.12. **Feladat** programja:

```
NewtonSysB := proc(B_0::Matrix(shape=square), F::Vector, x0::'Vector'(float),
                   kMAX::posint, eps1::positive, eps2::positive)
local n, k, x_k, x_k1, y_k, y_k1, B_k, B_k_inv, B_k1, F_k, F_k1, s_k, z_k;
uses LinearAlgebra;
Digits := myDigits;
n := RowDimension(B_0);
x_k := x0;
x_k1 := Vector(n);
B_k := evalf(B_0);
B_k_inv := MatrixInverse(B_k);
for k from 1 to kMAX do
    F_k:= subs({seq(x[j]=x_k[j], j=1..n)},F);
    s_k:= -B_k_inv.F_k;
    x_k1 := x_k + s_k;
    if (VectorNorm(F_k) < eps1) and (VectorNorm(x_k1 - x_k) < eps2) then
        return x_k1;
    else
        F_k1 := subs({seq(x[j]=x_k1[j], j=1..n)},F);
        y_k := F_k1 - F_k;
        B_k1 := B_k + 1/DotProduct(s_k,s_k)*(y_k-B_k.s_k).s_k^%T;
        x_k := x_k1;
        B_k := B_k1;
        F_k := F_k1;
        z_k := B_k_inv.y_k;
        B_k_inv := B_k_inv - (1/(s_k^%T.z_k))*(z_k - s_k).(s_k^%T.B_k_inv);
    end if;
end do;
return k, x_k1, "Nem konvergált az iteráció";
end proc:
## Teszteljük az eljárást:
#n := 2;
#myDigits := 15;
#F:=Vector([x[1]^2+x[2]^2-1, x[1]*x[2] - 2*x[1]^2+1]);
#J:=VectorCalculus[Jacobian](F,[x[1],x[2]]):
#solve({F[1], F[2]});
#evalf(allvalues(%));
#x0:=Vector([-0.84, -0.5]);
#B0 := subs({seq(x[j]=x0[j], j=1..n)},J);
#NewtonSysB(B0, F, x0, 10, 0.001, 0.001);
#x0:=Vector([-1., 1.]);
#B0 := subs({seq(x[j]=x0[j], j=1..n)},J);
#NewtonSysB(B0, F, x0, 10, 0.001, 0.001);
#x0:=Vector([-1., 1.]);
#B0 := subs({seq(x[j]=x0[j], j=1..n)},J);
#NewtonSysB(B0, F, x0, 10, 0.0000001, 0.0000001);
#F:=Vector([x[1]^2+x[2]^2-1, x[2] - 2*x[1]]);
#J:=VectorCalculus[Jacobian](F,[x[1],x[2]]):
#solve({F[1], F[2]});
#evalf(allvalues(%));
#x0:=Vector([0.5, 1.0]);
#B0 := subs({seq(x[j]=x0[j], j=1..n)},J);
#NewtonSysB(B0, F, x0, 20, 0.001, 0.001);
#x0:=Vector([-0.5, -1.0]);
#B0 := subs({seq(x[j]=x0[j], j=1..n)},J);
#NewtonSysB(B0, F, x0, 20, 0.001, 0.001);
#x0:=Vector([0.5, 0.25]);
#B0 := subs({seq(x[j]=x0[j], j=1..n)},J);
#NewtonSysB(B0, F, x0, 20, 0.001, 0.001);
#x0:=Vector([1.0, -1.0]);
#B0 := subs({seq(x[j]=x0[j], j=1..n)},J);
#NewtonSysB(B0, F, x0, 20, 0.001, 0.001);
```

A 10.2.5. **Feladat** programja:

```
OrtoPoliG := proc(a, b, rho, n)
local b_szorzat, p_list, a_list, i, j;
b_szorzat := proc(f, h)
return int(f*h*rho, x = a..b);
end proc:
p_list := [ 1 ];
for i from 1 to n do
a_list := [ seq( (-1)* b_szorzat(x^i, p_list[j])/b_szorzat(p_list[j], p_list[j]), j = 1..i )];
p_list := [ op(p_list),  x^i + add( a_list[j]*p_list[j], j = 1..i ) ];
end do;
return p_list[n+1];
end proc:

# A különböző paraméterek itt adhatók meg:
n := 3; a := -1; b := 1; rho := sqrt(1-x^2);
p_n := OrtoPoliG(a,b,rho,n);
# az alappontok NEM lesznek nagyság szerint rendezve!
x_list := [solve(p_n,x)]:
alappontok := map(Re,(evalf(x_list)));
w_list := [seq (int( p_n/((x-x_list[k])*subs( x=x_list[k], diff(p_n,x)))*rho, x=a..b),
    k=1..n)]:
sulyok :=  map(Re,(evalf(w_list)));

f := x -> x^2*abs(sin(x));
int(f(x)*rho, x= a..b);
evalf(%);
Gauss_int_pontos := sum ('w_list[j]'*'f(x_list[j])', j = 1..n);
evalf(%);
Gauss_int := add (sulyok[j]*f(alappontok[j]), j = 1..n);
```

A 10.2.6. **Feladat** programja:

```
restart;
n := 5;
h := 1/n*int(1,x=-1..1);
S := [seq ( n/2*int(x^k,x=-1..1), k = 1..n)];
Erendszer := [b[0] = 1, seq( S[k] + sum(S[k-j]*b[j], j=1..k-1) + k*b[k] = 0, k = 1..n )];
Mo        := solve( Erendszer, {seq('b[j]', j =0..n)} );
Mo_list   := map( rhs, convert(Mo,list) );
p         := PolynomialTools[FromCoefficientList]( ListTools[Reverse](Mo_list), x );
Gyokok    := allvalues( [solve(p)] ) :
simplify(Gyokok,radical);
evalf(Gyokok);
plot(p, x=-1..1, numpoints = 10000);
```

A 10.3.8. **Feladat** programja:

```
Romberg := proc(f, a::float, b::float, n::posint, reszletes::truefalse)
  local R, h, j, k;
  Digits := myDigits:
  R := array(0..n, 0..n):
  h := b - a:
  R[0,0] := h/2 * ( f(a)+f(b) ):
  for j from 1 to n do
    h := h/2:
    R[j,0] := 1/2*R[j-1,0] + sum(h*f(a+(2*m-1)*h), m=1..2^(j-1)):
    for k from 1 to j do
      R[j,k] := R[j,k-1] +(R[j,k-1] - R[j-1,k-1])/(4^k - 1):
    end do:
  end do:
  if (reszletes) then
    for j from 0 to n do
      for k from 0 to j do
        printf("%15.10f ",R[j,k]);
      end do:
      printf("\n");
    end do:
  end if:
  return(R[n,n]):
end proc:
## Az eljárás tesztelése
#myDigits := 20;
#a,b := 0,Pi;  f := x -> sin(x):
#Int(f(x),x=a..b)=int(f(x),x= a..b);plot(f(x),x=a..b);R_int:= Romberg(f,evalf(a),evalf(b),4,true);
#a,b := 0,1;  f := x -> exp(-x^2/2):
#Int(f(x),x=a..b)=int(f(x),x=a..b);plot(f(x),x=a..b);R_int:= Romberg(f,evalf(a),evalf(b),4,false);
#a,b := 0.01, 0.1;  f := x -> x*abs(sin(1/(x))):
#Int(f(x),x=a..b)=int(f(x),x=a..b);plot(f(x),x=a..b);R_int:= Romberg(f,evalf(a),evalf(b),4,true);
#a,b := -1,1;  f := x -> 1/(x^2+1):
#Int(f(x),x=a..b)=int(f(x),x=a..b);plot(f(x),x=a..b);R_int:= Romberg(f,evalf(a),evalf(b),4,false);
```

A 12.1.1. **Feladat** programja:

```
#include <stdio.h>
#include <gsl/gsl_linalg.h>

int
main (void) {
  double A_data[] = {
  1.,    3.,    3.,   -2.,
  1.,    2.,   -2.,   -1.,
  0.,    0.,    2.,    1.,
 -1.,   -3.,    1.,   -2.
  };
  double b_data[] = {
    -13.,
     -1.,
     -4.,
      5
    };
  gsl_matrix_view  A = gsl_matrix_view_array (A_data, 4, 4);
  gsl_vector_view  b = gsl_vector_view_array (b_data, 4);
  gsl_vector      *x = gsl_vector_alloc (4);
  gsl_permutation *p = gsl_permutation_alloc (4);
  int s;

  gsl_linalg_LU_decomp (&A.matrix, p, &s);
  gsl_linalg_LU_solve  (&A.matrix, p, &b.vector, x);

  printf ("\n  Az Ax = b egyenletrendszer megoldása: \n");
  printf ("\n  x* = \n");
  gsl_vector_fprintf (stdout, x, "%g");
  gsl_permutation_free (p);
  gsl_vector_free (x);
  return 0;
}
```

A 12.1.2. **Feladat** programja:

```
#include <stdio.h>
#include <gsl/gsl_linalg.h>

int
main (void) {
  double A_data[] = {
    1.,       3.,       3.,      -2.,
    1.,       2.,      -2.,      -1.,
    0.,       0.,       2.,       1.,
   -1.,      -3.,       1.,      -2.
  };
  gsl_matrix_view  A = gsl_matrix_view_array (A_data, 4, 4);
  gsl_vector*    tau = gsl_vector_alloc (4);
  gsl_matrix*      Q = gsl_matrix_alloc(4,4);
  gsl_matrix*      R = gsl_matrix_alloc(4,4);

  gsl_linalg_QR_decomp(&A.matrix,tau);
  gsl_linalg_QR_unpack(&A.matrix,tau,Q,R);

  printf ("\n  Az A = QR felbontás mátrixai: \n");
  printf ("\n  Q = \n");
  gsl_matrix_fprintf (stdout, Q, "%g");
  printf ("\n  R = \n");
  gsl_matrix_fprintf (stdout, R, "%g");
  gsl_vector_free (tau);
  gsl_matrix_free (Q);
  gsl_matrix_free (R);
  return 0;
}
```

A 12.1.3. **Feladat** programja:

```
#include <stdio.h>
#include <gsl/gsl_math.h>
#include <gsl/gsl_eigen.h>

int
main (void){
  double Hdata[] = {
    1.0,    1/2.0,  1/3.0,  1/4.0,
    1/2.0,  1/3.0,  1/4.0,  1/5.0,
    1/3.0,  1/4.0,  1/5.0,  1/6.0,
    1/4.0,  1/5.0,  1/6.0,  1/7.0
  };
  int i;
  gsl_matrix_view H    = gsl_matrix_view_array (Hdata, 4, 4);
  gsl_vector *eigenval = gsl_vector_alloc (4);
  gsl_matrix *eigenvec = gsl_matrix_alloc (4, 4);
  gsl_eigen_symmv_workspace * w = gsl_eigen_symmv_alloc (4);
  gsl_eigen_symmv (&H.matrix, eigenval, eigenvec, w);
  gsl_eigen_symmv_free (w);
  gsl_eigen_symmv_sort (eigenval, eigenvec, GSL_EIGEN_SORT_ABS_ASC);

  for (i = 0; i < 4; i++) {
    double eigenval_i = gsl_vector_get (eigenval, i);
    gsl_vector_view eigenvec_i = gsl_matrix_column (eigenvec, i);
    printf ("\n A(z) %d. sajátérték:\n\n%g\n", i+1, eigenval_i);
    printf ("\n A hozzá  tartozó sajátvektor: \n\n");
    gsl_vector_fprintf (stdout,&eigenvec_i.vector, "%g");
  }

  gsl_vector_free (eigenval);
  gsl_matrix_free (eigenvec);
  return 0;
}
```

A 12.1.4. **Feladat** programja:

```
include <stdlib.h>
#include <stdio.h>
#include <math.h>
#include <gsl/gsl_errno.h>
#include <gsl/gsl_spline.h>

int
main (void){
  int N = 5;
  double x[5] = {0.0, M_PI/2, M_PI, 3*M_PI/2, 2*M_PI};
  double y[5] = { sin(x[0]), sin(x[1]), sin(x[2]), sin(x[3]), sin(x[4]) };
  FILE * outfp;

  outfp = fopen("./gsl-cubic_spline.out", "w+");
  gsl_interp_accel *acc = gsl_interp_accel_alloc ();
  const gsl_interp_type *t = gsl_interp_cspline_periodic;
  gsl_spline *spline = gsl_spline_alloc (t, N);
  int i; double xi, yi;
  fprintf (outfp,"#m=0,S=5\n");
  for (i = 0; i < N; i++){
    fprintf (outfp,"%g %g\n", x[i], y[i]);
  }
  fprintf (outfp,"#m=1,S=0\n");
  gsl_spline_init (spline, x, y, N);
  for (i = 0; i <= 100; i++){
    xi = (1 - i / 100.0) * x[0] + (i / 100.0) * x[N-1];
    yi = gsl_spline_eval (spline, xi, acc);
    fprintf (outfp,"%g %g\n", xi, yi);
  }
  gsl_spline_free (spline);
  gsl_interp_accel_free (acc);
  fclose(outfp);
  return 0;
}
```

A 12.1.5. **Feladat** programja:

```
#include <stdio.h>
#include <gsl/gsl_math.h>
#include <gsl/gsl_chebyshev.h>

double
f (double x, void *p) {
  if (x < 0)
    return -1;
  else
   if (x = 0)
    return 0;
   else
    return 1;
}

int
main (void) {
  int i, n = 100;
  double a = -1;
  double b =  1;
  gsl_cheb_series *cs = gsl_cheb_alloc (40);
  gsl_function F;
  F.function = f;
  F.params = 0;
  gsl_cheb_init (cs, &F, a, b);

  printf ("\n Az f(x) és a cs_40(x) Csebisev-sorfejtés értéke az x_i = a + i*h pontokban:\n");
  printf ("\nx_i\tf(x_i)\tcs_40(x_i)\n");

  for (i = 0; i <= n; i++){
    double x = a + i*(b-a) / (double)n;
    double r40 = gsl_cheb_eval (cs, x);
    printf ("%g\t%g\t%g\n", x, GSL_FN_EVAL (&F, x),  r40);
  }
  gsl_cheb_free (cs);
  return 0;
}
```

A 12.1.6. Feladat programja:

```
#include <stdio.h>
#include <math.h>
#include <gsl/gsl_integration.h>

typedef double func (double x);

double f1 (double x) {
    double f = log(x)/sqrt(x);
    return f;
}

double f2 (double x) {
    double f = sin(1/x);
    return f;
}

double f3 (double x) {
    double f = 1/sqrt(1-x*x);
    return f;
}

void num_int(func *fx, double* result, double* error, double a, double b, double epsrel) {
    gsl_integration_workspace * w = gsl_integration_workspace_alloc (1000);

    gsl_function F;
    F.function = fx;
    F.params = NULL;

    gsl_integration_qags (&F, a, b, 0, epsrel, 1000, w, result, error);

    gsl_integration_workspace_free(w);
}

int main (void) {
    double result, error;

    num_int(&f1,&result,&error,0,1,1e-7);
    printf ("Az integrál közelítő értéke:\n\n% .18f\n\n", result);
    printf ("A becsült hiba:\n\n% .18f\n\n", error);

    num_int(&f2,&result,&error,0,M_PI_2,1e-5);
    printf ("Az integrál közelítő értéke:\n\n% .18f\n\n", result);
    printf ("A becsült hiba:\n\n% .18f\n\n", error);

    num_int(&f3,&result,&error,-1,1,1e-7);
    printf ("Az integrál közelítő értéke:\n\n% .18f\n\n", result);
    printf ("A becsült hiba:\n\n% .18f\n\n", error);

    return 0;
}
```

A 12.1.7. **Feladat** programja:

```
#include <stdio.h>
#include <f2c.h>
#include <clapack.h>

int
main( ) {
     char JOBU;
     char JOBVT;
     int i;
     integer N = 4;
     integer M = 4;
     integer LDA = 4;
     integer LDU = 4;
     integer LDVT = 4;
     integer LWORK;
     integer INFO;
     integer mn = 4;
     integer MN = 4;

     double A[4*4] = {
           2.0,  5.0,  3.0,  4.0,
           2.0,  1.0,  2.0,  4.0,
           3.0,  1.0,  6.0,  5.0,
           1.0,  8.0, 12.0,  1.0
       };
     double s[4];
     double wk[201];
     double uu[4*4];
     double vt[4*4];
     JOBU  = 'A';
     JOBVT = 'A';
     LWORK =  201;

    dgesvd_( &JOBU, &JOBVT, &M, &N, A, &LDA, s, uu, &LDU, vt, &LDVT, wk, &LWORK, &INFO);

    printf("\n Az A mátrix szinguláris értékei:\n\n");

    for ( i= 0; i< N; i++ ) {
        printf("s[%d] = %f\n", i+1, s[ i ] );
    }
    return 0;
}
```

A 12.1.8. **Feladat** programja:

```
restart;
with (LinearAlgebra):
N := 4; Ainp1 := Matrix([[[2,-2,0,-1],[1,0,0,0],[0,-1,2,-2],[-2,-2,1,-1]], datatype=float[8]);
A := Copy(Ainp1):

# A DGEEV külső eljárás deklarálása a CLAPACK kézikönyv szerinti paraméterekkel
my_DGEEV := define_external(
  "dgeev_",
  LIB    = "libclapack.so",
  'JOBVL'::REF(char[1]),    # V, ha kellenek a baloldali sajátvektorok, N, ha nem (input)
  'JOBVR'::REF(char[1]),    # V, ha kellenek a jobboldali sajátvektorok, N, ha nem (input)
  'N'    ::REF(integer[4]), # A mátrix dimenziója (input)
  'A'    ::ARRAY(float[8]), #  mátrix (input)
  'LDA'  ::REF(integer[4]), # A fődimenziója (input)
  'WR'   ::ARRAY(float[8]), # a sajátértékek valós részeiből álló vektor (output)
  'WI'   ::ARRAY(float[8]), # a sajátértékek képzetes részeiből álló vektor (output)
  'VL'   ::ARRAY(float[8]), # baloldali sajátvektorok mátrixa (output)
  'LVDL' ::REF(integer[4]), # VL fődimenziója
  'VR'   ::ARRAY(float[8]), # jobboldali sajátértékek mátrixa (output)
  'LVDR' ::REF(integer[4]), # VR fődimenziója
  'WORK' ::ARRAY(float[8]), # munkaváltozó
  'LWORK'::REF(integer[4]), # munkaváltozó
  'INFO' ::REF(integer[4])  # info
):

JOBVL := "N":
JOBVR := "V":
LDA   := N:
WR    := Vector(N,datatype=float[8]):
WI    := Vector(N,datatype=float[8]):
VL    := Matrix(N,N,datatype=float[8]):
LDVL  := N:
VR    := Matrix(N,N,datatype=float[8]):
LDVR  := N:
WORK  := Matrix(4*N, 4*N,datatype=float[8]):
LWORK := 4*N:
INFO  := 0:

my_DGEEV(JOBVL,JOBVR,N,A,LDA,WR,WI,VL,LDVL,VR,LDVR,WORK,LWORK,INFO):
# A CLAPACK által kiszámolt eredmények:
'WR' = WR;
'WI' = WI;
'VR' = VR;
# A Maple által kiszámolt eredmények:
Eigenvectors(A);
```

Irodalomjegyzék

[1] N. Sz. Bahvalov, A gépi matematika numerikus módszerei, Műszaki Könyvkiadó, Budapest, 1977.

[2] Csendes Tibor, Közelítő és szimbolikus számítások, Polygon Jegyzettár, Szeged, 2007.

[3] J. Douglas Faires and Richard Burden, Numerical Methods, Third Edition, Brooks/Cole, 2003.

[4] Hartung Ferenc, Bevezetés a numerikus analízisbe, Veszprémi Egyetemi Kiadó, Veszprém, 2004

[5] John H. Mathews and Kurtis K. Fink, Numerical Methods Using Matlab, 4th Ed., Prentice Hall, 2004.

[6] Molnárka Győző, Gergó László, Wettl Ferenc, Horváth Attila, Kallós Ferenc, A Maple V és alkalmazásai, Springer Hungarica Kiadó Kft., Budapest, 1996.

[7] Móricz Ferenc, Numerikus módszerek az algebrában és analízisben, Polygon Jegyzettár, Szeged, 1997.

[8] Móricz Ferenc, Differenciálegyenletek numerikus módszerei, Polygon Jegyzettár, Szeged, 1998.

[9] Móricz Ferenc, Bevezetés a numerikus matematikába, Polygon Jegyzettár, Szeged, 2008.

[10] I. P. Natanszon, Konstruktív függvénytan, Akadémiai Könyvkiadó, Budapest, 1952.

[11] R. Plato, Concise numerical mathematics, American Mathematical Society, 2003.

[12] W. Rudin, A matematikai analízis alapjai, Műszaki Könyvkiadó, Budapest, 1978.

[13] J. Stoer and R. Bulirsch, Introduction to numerical analysis, Springer Verlag, New York, 3rd Ed. 2002.

[14] Stoyan Gisbert, Takó Galina: Numerikus módszerek I-II.,ELTE-TypoTeX, Budapest,1993,1995.

[15] Stoyan Gisbert (szerk.) Matlab – frissített kiadás, Typotex Kiadó, 2005.

[16] Szendrei Ágnes, Diszkrét matematika, Polygon Könyvtár, Szeged, 1995.

[17] Virágh János, Numerikus Matematika, JATEPress, Szeged, 1997.

[18] E. Anderson & al., LAPACK User's Guide, 3rd Ed., SIAM, Philadelphia, 1999

[19] M. Galassi & al., GNU Scientific Library Reference Manual - 3rd Ed. (v1.12), Network Theory Limited, 2009.