Eötvös Loránd Tudományegyetem

Természettudományi Kar

Rózsa Petra

Szimmetrikus, pozitív definit együtthatómátrixú lineáris algebrai egyenletrendszerek iteratív megoldási módszerei

BSc Elemző Matematikus Szakdolgozat

Témavezető:

Fekete Imre
Alkalmazott Analízis és Számításmatematikai Tanszék

Budapest
2018

Köszönetnyilvánítás

Elsősorban szeretném megköszönni témavezetőmnek, Fekete Imrének, lelkiismeretes munkáját, idejét. Magas fokú szakértelmével végig segítette munkámat.

Továbbá szeretném megköszöni páromnak, akinek segítsége nélkül nem jutottam volna el idáig.

Tartalomjegyzék

1. fejezet

Bevezetés

Matematikai tanulmányaink során lépten-nyomon lineáris algebrai egyenletekre bukkanunk. Kezdve az általános iskolai, középiskolai éveinken át egyetemi éveinkig. Egyre kifinomultabb eljárásokat és technikákat ismerünk meg annak érdekében, hogy egyre nagyobb méretű lineáris algebrai egyenletrendszer (LAER) megoldását találjuk meg. Tömören ezeket a LAER-eket

$$Ax = b$$

mátrix-vektor alakban tekintjük, ahol az $A \in \mathbb{R}^{n \times n}$ mátrix és $b \in \mathbb{R}^n$ vektor adott, míg $x \in \mathbb{R}^n$ a keresendő megoldó vektor. Tipikusan ezek a rendszerek jellemezhetnek különböző gazdasági, pénzügyi, mérnöki modellek ismeretlenjeit. Bevezető algebra kurzuson alapvető megoldhatósági, míg numerikus lineáris algebra és numerikus módszerek kurzusokon direkt és iteratív módszerekkel ismerkedünk meg annak érdekében, hogy a megoldást meghatározzuk. Ezeket kis n értékek esetén kézzel pontosan is meg tudjuk határozni. Azonban nagy n értékek esetén számítógépet kell használnunk, s így a gépi matematikai alapismereteinkre alapozva a megoldást csupán kellő pontossággal szeretnénk megadni. Mielőtt pontosan elmerülnénk ezekben a módszerekben álljon előttünk példaként, hogy az emberiség mekkora méretű egyenletrendszerek megoldásával birkózott meg az első nagy teljesítményű szuperszámítógépek segítségével. Ezeket az 1.1 Táblázatban foglaltuk össze.

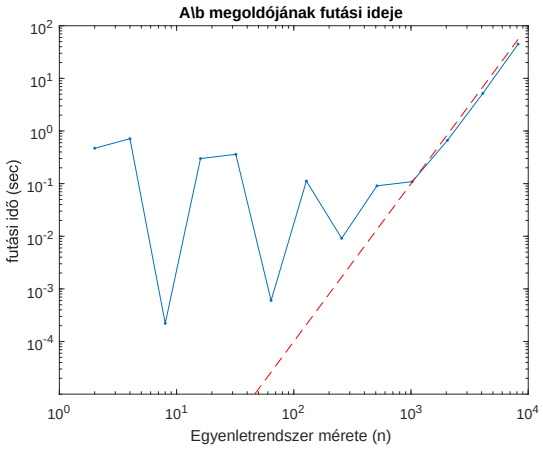
Az 1.1 Táblázat azt mutatja, hogy a kezdeti 1950-es 20×20-as méretű LAER-ek helyett ma már 200000×200000-es méretű LAER-eket is meg lehet oldani. Azaz bő 60 év alatt 10^4 nagyságrenddel nagyobb LAER-ket tudunk szuperszámítógépek segítségével megoldani. Kezdetben egy szuperszámítógép körülbelül 1 FLOPS (Floating Point Opera-

Év	n
1950	20
1965	200
1980	2000
1995	20000
2010	200000

1.1. táblázat. Átlagos $n \times n$-es méretű LAER méret átlagos szuperszámítógépes megoldás esetén.

tions Per Second) lebegőpontos műveletet végzett másodpercenként. Ma egy átlagos szuperszámítógép már 10^{12} FLOPS-ra képes, azaz másodpercenként billió osztási, szorzási, kivonási és összeadási operációs műveletre képes. Ez bő 60 év alatt 10^{12} nagyságrend javulást jelent. Érdekességként megjegyeznénk, hogy magyar szuperszámítógép a TOP500-as félévente frissülő szuperszámítógépes listán 2015. novemberében szerepelt utoljára (Név: Leó NIIFI-Debrecen, Pozíció: 401, $253.6 \cdot 10^{12}$ FLOPS, részletekért lásd [9]). De mit is jelentenek ezek az adatok?

Az adatok $10^{12}/10^4 = 10^3$ hányadosa az algoritmikus $\mathcal{O}(n^3)$ korlátot mutatja a technológiai fejlődés függvényében, azaz a bevezető órákon tanult klasszikus Gauss-elimináció vagy LU-felbontás műveletigényeire reflektálnak. Ehhez tekintsünk egy egyszerű MATLAB futtatás eredményét (részleteket mellőzve) a LAER méretét növelve a futási idő függvényében. Az eredményeinket log-log alapon az 1.1 Ábra foglalja össze.



1.1. ábra. A népszerű MATLAB szoftver beépített direkt megoldájának futási sebessége adott méretű LAER esetén.

Az 1.1 Ábra szaggatott piros vonala a log-log ábrához tartozó n^3. Azaz mini példánk is mutatja az $\mathcal{O}(n^3)$ algoritmikus korlátot tetszőleges LAER esetén. A szuperszámítógépek a nagyobb méretű rendszerek megoldásában segítenek. Természetesen felmerülő kérdés, hogy ezt az algoritmikus korlátot milyen speciális struktúrájú LAER-ek esetén tudjuk áttörni. A szakdolgozatban *szimmetrikus, pozitív definit* együtthatómárixú LAER-ekkel és azok iteratív megoldási módszereivel foglalkozunk majd. Mivel szuperszámítógép nem áll rendelkezésünkre, ezért elméleti eredményeink megadásán túl kísérleteinket egy átlagos laptop segítségével hajtjuk végre. A szakdolgozat az egyes fejezeteiben az alábbi dolgokra fókuszálnak:

- A 2. Fejezetben klasszikus iteratív módszerek konstruálásával és azokhoz tartozó konvergenciatételekkel ismerkedünk meg. Különböző gyakorlatban felmerülő példák megoldása során az implementálási kérdések fontosságát is érzékeltetjük az egyes módszerek összehasonlítása mellett.

- A 3. Fejezetben a mai napig a leggyakrabban használt Krylov-altér alapvető iterációs eljárásával foglalkozunk elméleti és gyakorlati tekintetben egyaránt. A jobb megértéshez és a demonstrálás okán példákat is mutatunk.

2. fejezet

Klasszikus iterációs módszerek

Ahogy az 1. Fejezetben is említettük, lineáris algebrai egyenletrendszer (LAER) megoldását direkt eljárások mellett iteratív módon is megkaphatjuk. Ehhez tekintsük az

$$Ax = b \tag{2.1}$$

alakot, ahol az $A \in \mathbb{R}^{n\times n}$ mátrix és a $b \in \mathbb{R}^n$ vektor adottak. Korábbi ismereteinkre hagyatkozva a (2.1) LAER $x \in \mathbb{R}^n$ megoldása egyértelmű, ha az A mátrix inverze, azaz A^{-1} létezik. A (2.1) LAER-t koordinátás alakban felírva kapjuk az

$$a_{i1}x_1 + a_{i2}x_2 + \ldots + a_{ii}x_i + \ldots + a_{in}x_n = b_i$$

formát minden $i = 1, \ldots, n$ esetén. Tegyük fel, hogy az A mátrix diagonálisában egyetlen elem sem nulla, azaz tetszőleges $i = 1, \ldots, n$ esetén $a_{ii} \neq 0$. Ekkor a fenti koordinátás alakot x_i-re rendezve kapjuk, hogy

$$x_i = \frac{b_i}{a_{ii}} - \sum_{\substack{j=1 \\ j\neq i}}^{n} \frac{a_{ij}}{a_{ii}}x_j = \frac{b_i}{a_{ii}} - \sum_{j=1}^{i-1} \frac{a_{ij}}{a_{ii}}x_j - \sum_{j=i+1}^{n} \frac{a_{ij}}{a_{ii}}x_j, \quad i = 1, \ldots, n. \tag{2.2}$$

Jelölje az x_i^k az x megoldóvektor i-edik koordinátájának k-adik komponensét. Ekkor a (2.2) egyenlet bal oldala lesz az új, $k+1$-edik iterált érték, míg az egyenlet jobb oldalán szereplő x_j megoldókomponens iterált értékének megválasztására több lehetőség is adódik. Ha mind a két tagban a k-adik iterált értékkel számolunk, akkor az alábbi

$$x_i^{k+1} = \frac{b_i}{a_{ii}} - \sum_{j=1}^{i-1} \frac{a_{ij}}{a_{ii}}x_j^k - \sum_{j=i+1}^{n} \frac{a_{ij}}{a_{ii}}x_j^k, \quad i = 1, \ldots, n, \ k = 0, \ldots, K \tag{2.3}$$

összefüggéshez jutunk. A fenti (2.3) módszer az ún. *Jacobi iteráció*. A módszer elindításához adott x_i^0 értékek megadására van szükségünk minden i index esetén. A tömör és kényelmes használat érdekében mátrix-vektor alakban szeretnék megadni a Jacobi iterációt. Ehhez vezessük be az A mátrix

$$A = L + D + U \tag{2.4}$$

felbontását, ahol L szigorú alsó háromszög, D diagonál, míg U szigorú felső háromszög mátrixa A-nak. Ekkor a (2.3) koordinitás alak a fenti (2.4) felbontásban szereplő mátrixok segítségével tömören az

$$x^{k+1} = D^{-1}b - D^{-1}Lx^k - D^{-1}Ux^k = -D^{-1}(L+U)x^k + D^{-1}b \tag{2.5}$$

alakban írható fel. A Jacobi iteráció megindításához pedig egy adott x^0 kezdeti vektor megadására van szükségünk. A fenti mátrix-vektor alak a (2.4) felbontással közvetlenül származtatható. Ugyanis

$$\begin{aligned} (L+D+U)x &= b \\ Dx &= b - (L+U)x \\ x &= D^{-1}b - D^{-1}(L+U)x \\ x &= -D^{-1}(L+U)x + D^{-1}b \end{aligned}$$

Azaz a (2.5) egyenlet

$$x^{k+1} = -D^{-1}(L+U)x^k + D^{-1}b$$

alakját kaptuk vissza.

Ha a Jacobi iteráció megalkotásához használt ötletünkkel szemben a (2.3) alak jobboldalában az x-et tartalmazó első tagban a $k+1$-edik, a második tagban a k-adik iterált értékkel dolgozunk, akkor a (2.2) egyenletre nyert iterációs képlet az

$$x_i^{k+1} = \frac{b_i}{a_{ii}} - \sum_{j=1}^{i-1} \frac{a_{ij}}{a_{ii}} x_j^{k+1} - \sum_{j=i+1}^{n} \frac{a_{ij}}{a_{ii}} x_j^k, \quad i = 1, \ldots, n, \;\; k = 0, \ldots, K \tag{2.6}$$

alakot ölti. Ez az ún. *Gauss-Seidel iteráció*. A Jacobi iterációhoz hasonló gondolatot használva a módszer mátrix-vektor alakja

$$\begin{aligned} x^{k+1} &= D^{-1}b - D^{-1}Lx^{k+1} - D^{-1}Ux^k \\ Dx^{k+1} &= b - Lx^{k+1} - Ux^k \\ (L+D)x^{k+1} &= -Ux^k + b \\ x^{k+1} &= (L+D)^{-1}\left(-Ux^k + b\right) \end{aligned} \tag{2.7}$$

A módszer megindításához ismételten egy x^0 kezdeti vektorra van szükségünk. Ismételten a (2.4) felbontást használva közvetlenül számolhatjuk a fenti alakot. Ugyanis

$$\begin{aligned}(L+D+U)x &= b\\ (L+D)x &= b-Ux\\ x &= (L+D)^{-1}(-Ux+b)\end{aligned}$$

Azaz a (2.7) egyenlet

$$x^{k+1} = (L+D)^{-1}\left(-Ux^k+b\right)$$

alakját kaptuk vissza. Természetesen felmerülő kérdés, hogy miért így választottuk az iterált értékek indexét. A (2.6) felírást nézve azt figyelhetjük meg, hogy az új iterált érték i-komponensének számításához az új iterált érték korábbi komponenseit azonnal felhasználjuk. Könnyen meggondolható az indexek fordított megválasztása nem lenne célszerű. A Gauss-Seidel iterációtól intuitív módon kevesebb számítást várunk el, hiszen a Jabobi iteráció a régi iterált érték összes komponensével dolgozott.

2.0.1. Példa. Tekintsük azt a LAER-t, ahol A és b alakja az alábbi:

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

A fenti LAER-re a Jacobi és a Gauss-Seidel iterációkat alkalmazzuk $x^0 = (1\ 1\ 1)^T$ kezdeti vektorral és az adott iterációkkal egy-egy lépést teszünk meg. A Jacobi iteráció (2.3) koordinátás alakjával számolva az

$$\begin{aligned}x_1^1 &= \frac{1}{2}(1-(-1\cdot 1+0\cdot 1)) = 1\\ x_2^1 &= \frac{1}{2}(2-(-1\cdot 1-1\cdot 1)) = 2\\ x_3^1 &= \frac{1}{2}(3-(0\cdot 1-1\cdot 1)) = 2\end{aligned}$$

komponenseket kapjuk, azaz az $x^1 = (1\ 2\ 2)^T$ vektort. A Gauss-Seidel iteráció (2.6) koordinátás alakjával az

$$\begin{aligned}x_1^1 &= \frac{1}{2}(1-(-1\cdot 1+0\cdot 1)) = 1\\ x_2^1 &= \frac{1}{2}(2-(-1\cdot 1-1\cdot 1) = 2\\ x_3^1 &= \frac{1}{2}(3-(0\cdot 1-1\cdot 2)) = \frac{5}{2}\end{aligned}$$

azaz az $x^1 = (1\ 2\ 5/2)^T$ vektort kapjuk. ♣

2.1. Egylépéses stacionárius iterációk és konvergencia

Az előző alfejezetben tárgyalt Jacobi és Gauss-Seidel iterációs módszerek egyszerű algebrai átalakítások után átírhatóak közel hasonló alakban. A Jacobi iteráció a

$$D(x^{k+1} - x^k) + Ax^k = b \tag{2.8}$$

alakban, míg a Gauss-Seidel iteráció az

$$(L + D)(x^{k+1} - x^k) + Ax^k = b \tag{2.9}$$

alakban. A fenti (2.8)-(2.9) felírások egy általános egylépéses iteratív felírási alakot motiválnak, mégpedig a

$$B(x^{k+1} - x^k) + Ax^k = b$$

alakot, ahol $B \in \mathbb{R}^{n\times n}$ mátrix. Természetes általánosításaként pedig a

$$B_{k+1}\frac{x^{k+1} - x^k}{\omega_{k+1}} + Ax^k = b \tag{2.10}$$

ún. *egylépéses instacionárius iterációs* alakot adhatjuk meg. Az egylépéses iteráció a (2.10) alakban magától értetődő, hiszen a $k+1$-edik iterált értéket a k-adik iterált érték segítségével nyerjük. Az instacionárius szó pedig arra utal, hogy mind a $B_{k+1} \in \mathbb{R}^{n\times n}$ mátrix, mind a $\omega_{k+1} \in \mathbb{R}^+$ paraméter iterációról iterációra változhat. Ha ezek az iterációs eljárás során nem változnak, akkor ún. *stacionárius iteráció*ról beszélünk. Ekkor a (2.10) alak a

$$B\frac{x^{k+1} - x^k}{\omega} + Ax^k = b \tag{2.11}$$

alakra egyszerűsödik. Ahogy később látni fogjuk az ω pozitív paraméter célja egyes módszerek esetén az esetleges konvergencia gyorsítás. A fenti (2.11) alakban az eddig tanulmányozott módszerek az alábbi megválasztással nyerhetőek:

- ◇ Jacobi iteráció: $B = D$, $\omega = 1$,
- ◇ Gauss-Seidel iteráció: $B = L + D$, $\omega = 1$.

A Gauss-Seidel iteráció természetes általánosításaként a

$$(D + \omega L)\frac{x^{k+1} - x^k}{\omega} + Ax^k = b \tag{2.12}$$

alakot nyerjük. Ha $\omega > 1$, akkor a módszer *túlrelaxált* (overrelaxation), egyébként pedig *alulrelaxált*. A (2.12) módszerben az $\omega = 1$ megválasztás a már ismert Gauss-Sediel

iterációt eredményezi. Mindezekből adódóan a fenti (2.12) iterációs módszer az ún. *SOR iteráció* (successive overrelaxation).

Természetes kérdés, hogy a (2.11) stacionárius iteráció eredményeként nyert x^{k+1} numerikus megoldóvektor milyen feltételek mellett tart a (2.1) LAER x^* megoldásához. Ennek első lépéseként írjuk be az x^* megoldás vektort a stacionárius iteráció (2.11) alakjába, azaz

$$B\frac{x^* - x^*}{\omega} + Ax^* = b. \tag{2.13}$$

A (2.13) helyettesítés eredményeként pontosan oldjuk meg az eredeti LAER-t. Ezért célszerű a numerikus és pontos megoldás különbségét, azaz a hibát vizsgálni. Ehhez vezessük be az

$$e^k = x^k - x^*$$

jelölést. Az $e^k \in \mathbb{R}^n$ vektor neve az ún. *hibavektor*. Ilyen definíció mellett a (2.11) és (2.13) egyenletek különbségére adódó

$$B\frac{e^{k+1} - e^k}{\omega} + Ae^k = 0 \tag{2.14}$$

ún. *hibaegyenlet* adódik. Az iterációs eljárás jóságát az jelentené, ha az e^k hibavektor minden egyes iteráció után egyre inkább közeledne az azonosan nulla vektorhoz. Formálisan fogalmazva a

$$\lim_{k\to\infty} e^k = 0$$

feltétel teljesülését várjuk el. Mielőtt kimondjuk és bizonyítjuk a (2.11) stacionárius iterációra vonatkozó konvergencia tételt definiáljuk a tétel feltételeiben használandó fogalmakat.

2.1.1. Definíció. *Egy $A \in \mathbb{R}^{n\times n}$ mátrixot szimmetrikusnak nevezünk, ha $A^T = A$.*

2.1.2. Definíció. *Egy $A \in \mathbb{R}^{n\times n}$ mátrixot pozitív definitnek nevezünk, ha $(Ax, x) > 0$ tetszőleges $x \in \mathbb{R}^n \setminus \{0\}$ esetén.*

2.1.3. Megjegyzés. Ha $A \in \mathbb{R}^{n\times n}$ egy szimmetrikus pozitív definit mátrix, akkor létezik $\delta > 0$, hogy $(Ax, x) \geq \delta||x||^2$ tetszőleges $x \in \mathbb{R}^n \setminus \{0\}$ esetén. A δ az A mátrix legkisebb sajátértékének választható, amely egy pozitív valós szám.

2.1.4. Tétel. *[Egylépéses stacionárius iteráció konvergenciája] Tegyük fel, hogy A szimmetrikus pozitív definit mátrix és a*

$$B - 0.5\omega A$$

mátrix pozitív definit. Ekkor a (2.11) stacionárius iteráció konvergens.

Bizonyítás. Induljunk ki a (2.14) hibaegyenletből és rendezzük azt a Be^{k+1}-es tagjára. Ekkor

$$Be^{k+1} = \left(Be^k - \omega Ae^k\right) = (B - \omega A)e^k.$$

A fenti egyenletből az

$$e^{k+1} = (I - \omega B^{-1}A)e^k$$

és

$$Ae^{k+1} = (A - \omega AB^{-1}A)e^k$$

összefüggések közvetlenül adódnak. Tekintsük az (Ae^{k+1}, e^{k+1}) skalárszorzatot és használjuk a most nyert összefüggéseinket. Ekkor

$$\left(Ae^{k+1}, e^{k+1}\right) = \left(Ae^k - \omega AB^{-1}Ae^k, e^k - \omega B^{-1}Ae^k\right).$$

A fenti skalárszorzatot kibontva kapjuk, hogy

$$\left(Ae^{k+1}, e^{k+1}\right) = \left(Ae^k, e^k\right) - \omega\left(AB^{-1}Ae^k, e^k\right) - \omega\left(Ae^k, B^{-1}Ae^k\right) + \omega^2\left(AB^{-1}Ae^k, B^{-1}Ae^k\right).$$

Az A mátrix szimmetriáját és a tér valós vektortér tulajdonságát felhasználva az

$$\left(AB^{-1}Ae^k, e^k\right) = \left(B^{-1}Ae^k, Ae^k\right) = \left(Ae^k, B^{-1}Ae^k\right)$$

alakra jutunk. Ekkor

$$\left(Ae^{k+1}, e^{k+1}\right) = \left(Ae^k, e^k\right) - 2\omega\left(Ae^k, B^{-1}Ae^k\right) + \omega^2\left(AB^{-1}Ae^k, B^{-1}Ae^k\right).$$

Vezessük be a $J^k = \left(Ae^k, e^k\right)$ és $y^k = B^{-1}Ae^k$ jelöléseket. Utóbbiból adódik, hogy $Ae^k = By^k$. Ekkor a teljes skalárszorzatra a bevezetett jelöléseink mellett kapjuk, hogy

$$J^{k+1} = J^k - 2\omega\left(By^k, y^k\right) + \omega^2\left(Ay^k, y^k\right).$$

Nyilvánvaló összevonás után

$$J^{k+1} = J^k - 2\omega\left[\left(By^k, y^k\right) - 0.5\omega\left(Ay^k, y^k\right)\right] = J^k - 2\omega\left(\left(B - 0.5\omega A\right)y^k, y^k\right). \quad (2.15)$$

Mit mondhatunk el a $\left(J^{k+1}\right) \subset \mathbb{R}^n$ sorozatról?

Tetszőleges $x^0 \in \mathbb{R}^n$ kezdeti vektor mellett

- ◇ a 2.1.3 Definíció értelmében $J^k = \left(Ae^k, e^k\right) \geq 0$,
- ◇ a $B - 0.5\omega A$ mátrixra vonatkozó pozitív definit feltétel garantálja, hogy a (2.15) egyenlet $-2\omega \left(\left(B - 0.5\omega A\right) y^k, y^k\right)$ tagja nempozitív.

Azaz a $\left(J^{k+1}\right) \subset \mathbb{R}^n$ sorozat alulról korlátos és monoton csökkenő sorozat. Ekkor pedig a sorozat konvergens, azaz

$$\lim_{k\to\infty} J^{k+1} = J^*.$$

Továbbá ismételten a $B - 0.5\omega A$ mátrixra vonatkozó pozitív definit feltételt felhasználva

$$\lim_{k\to\infty} y^k = 0.$$

Mivel $e^k = A^{-1}By^k$, ezért

$$||e^k|| \leq ||A^{-1}B|| \cdot ||y^k||,$$

ahol az $||A^{-1}B||$ mátrixnorma k értékétől függetlenül korlátos marad, ezért

$$\lim_{k\to\infty} e^k = 0,$$

azaz a (2.11) konvergens. □

2.1.1. Specifikus tételek

Ebben a fejezetben a dolgozat szempontjából releváns állításokat közlünk, melyek közül egy speciális állítást igazolunk.

2.1.5. Tétel. *[SOR iteráció konvergencia szükséges feltétele] Tegyük fel, hogy A szimmetrikus pozitív definit mátrix. Ekkor a SOR iteráció konvergens, ha $\omega \in (0, 2)$.*

Bizonyítás. *Induljunk ki a SOR iteráció (2.12) alakjából. Mivel az A mátrix szimmetrikus, ezért a (2.4) felbontás $A = L + D + L^T$ alakban írható fel. Tekintsük az (Ax, x) skalárszorzatot. Kibontva, szimmetriát és a valós vektorteret használva az*

$$\begin{aligned}(Ax, x) &= (Lx, x) + (Dx, x) + (L^Tx, x) = (Lx, x) + (Dx, x) + (x, Lx)\\ &= (Lx, x) + (Dx, x) + (Lx, x) = 2(Lx, x) + (D, x)\end{aligned}$$

összefüggéshez jutunk. A 2.1.4 Tétel $B - 0.5\omega A$ pozitív definit feltételét szeretnénk ellenőrizni. A tételben szereplő B mátrix ebben az esetben az $D + \omega L$ mátrix. Ekkor a fenti

(Ax, x) skalárszorzatra kapott eredményt felhasználva kapjuk, hogy

$$\begin{aligned}(Bx, x) - (0.5\omega Ax, x) &= ((D + \omega L)x, x) - 0.5\omega(Ax, x) \\ &= (Dx, x) + \omega(Lx, x) - 0.5\omega(2(Lx, x) + (D, x)) \\ &= (1 - 0.5\omega)(Dx, x).\end{aligned}$$

Mivel a fenti (Dx, x) skalárszorzat nemnegatív, így a pozitív definitséghez az $1 - 0.5\omega > 0$ feltételnek kell teljesülnie. Ez pedig pontosan akkor teljesül, ha $\omega \in (0, 2)$. □

Jól látható módon a fenti 2.1.5 Tétel a 2.1.4 Tétel feltételrendszerén gyengített. Ugyanakkor, hogy az $\omega \in (0, 2)$ feltétel elégséges is, azt már más technikával szokásos bizonyítani. Ennek első lépése, hogy az egylépéses stacionárius iterációkat szokás

$$x^{k+1} = Hx^k + v \tag{2.16}$$

alakban is írni. Ekkor érdemes az eredeti A mátrix egy alkalmas felbontását vizsgálni.

2.1.6. Definíció. *Az $A \in \mathbb{R}^{n\times n}$ mátrix $A = M - N$ felbontását reguláris felbontásnak nevezzük, ha $M^{-1} \geq 0$ és $N \geq 0$.*

Ilyen definíció mellett a fenti (2.16) iteráció

$$x^{k+1} = M^{-1}Nx^k + M^{-1}b \tag{2.17}$$

alakban írható, azaz $H = M^{-1}N$ és $v = M^{-1}b$. Továbbá a (2.4) helyett

$$A = D - \tilde{L} - \tilde{U} \tag{2.18}$$

felbontással dolgoznak, ahol $\tilde{L} = -L$ és $\tilde{U} = -U$. Ekkor a SOR iteráció reguláris felbontásához (2.18) alapján az

$$M = \frac{1}{\omega}(D - \omega L) \quad \text{és} \quad N = \frac{1}{\omega}\left[(1 - \omega)D + \omega U\right]$$

mátrixokat kapjuk, míg

$$H_\omega = M^{-1}N = (D - \omega L)^{-1}\left[(1 - \omega)D + \omega U\right].$$

Ekkor a szakirodalomban ma már csak Ostrowski-Reich tétel az alábbi eredményt állítja.

2.1.7. Tétel. (Ostrowski-Reich tétel) *Ha egy $A \in \mathbb{R}^{n\times n}$ mátrix szimmetrikus és pozitív diagonálisú, akkor a SOR iteráció $\omega \in (0, 2)$ esetén pontosan akkor konvergál, ha A pozitív definit.*

Bizonyítás. Az eredeti bizonyítás a [5] cikkben található. □

2.1.8. Következmény. Ha A szimmetrikus, pozitív definit mátrix, akkor a Gauss-Seidel iteráció tetszőleges kezdővektor mellett konvergens.

2.1.9. Megjegyzés. A 2.1.7 Tétel bizonyítása a $\rho(H_\omega) < 1$ kritériumon alapszik, azaz a SOR módszerhez társított reguláris felbontás spektrálsugarának nagyságán. Érdekes még megjegyezni, hogy Ostrowski nevéhez fűződik a SOR iterációra vonatkozó 1954-es eredmény [5]. A bizonyítása Reich 1949-es Gauss-Seidel eredményén ($\omega = 1$) alapszik [6], s ezért is említik a teljes tételt Ostrowski-Reich tételként.

2.2. Numerikus példák és programozási megjegyzések

Az előző alfejezetben látott tételekben szereplő iterációkat (Jacobi, Gauss-Seidel, SOR) szeretnénk gyakorlatban is alkalmazni. Ez az iterációk programozásán túl alkalmas leállási feltételek beépítését is jelentik. Ebben az alfejezetben fokozatosan egyes példák során szeretnénk a tanult módszereket érdemben összehasonlítani. Első lépésként algoritmikusan reprodukáljuk a 2.0.1. Példa feladatát.

2.2.1. Példa. A begépelésen túl (2.16) alakban felírva (2.4) felbontással dolgozva írunk erre a speciális LAER-re egy k iteráció után leálló programot.

```
A=[2 -1 0; -1 2 -1; 0 -1 2];
b=[1 2 3]';
x0=[1 1 1]';
k=1; %iteraciok szama
D=diag(diag(A));
L=tril(A, -1);
U=triu(A, 1);
% Jacobi iteracio
x_k_J=x0;
H_J=eye(size(A,1))-D\A;
v_J=D\b;
for i=1:k
    x_k_J=H_J*x_k_J+v_J;
end
```

```
15 x_k_J
16 % Gauss-Seidel iteracio
17 x_k_GS=x0;
18 H_GS=-(L+D)\U;
19 v_GS=(L+D)\b;
20 for i=1:k
21     x_k_GS=H_GS*x_k_GS+v_GS;
22 end
23 x_k_GS
```

A futás eredményeként az

```
x_k_J =

   1
   2
   2

x_k_GS =

   1.0000
   2.0000
   2.5000
```

vektorokat kapjuk. Mivel a program tetszőleges iterációszám mellett működik, ezért nézzük meg, hogy a $k = 20$ iterációs választás mellett a két iteráció eljárás eredménye mennyire közelíti a feladat pontos $x = (2.5\ 4\ 3.5)^T$ megoldását. A $k = 20$ választás mellett az

```
x_k_J =

   2.4375
   3.9062
   3.4375

x_k_GS =
```

```
2.4961
3.9961
3.4980
```

eredményeket kapjuk, melyek közel vannak a feladat pontos megoldásához. ♣

A fenti 2.2.1 Példában a k paraméter megválasztása a maximális iterációs lépés leállási feltételét jelenti. További természetes elvárás lehet egy adott programmal szemben, hogy biztonságos módon, előírt feltételek mellett álljon le. Ilyen lehet a mostani programunkba beépített

◇ Maximális iterációs szám

$$maxit$$

Ahhoz, hogy a korábban tanult feltételek mellett és tételek alapján a konvergens iterációs eljárásaink kívánt hibahatáron belül – azaz másképpen fogalmazva adott toleranciával (TOL) – álljanak le arra tipikusan 3 lehetőség adott.

◇ Abszolút hiba adott toleranciával

$$||x^{k+1} - x^k|| < TOL$$

◇ Relatív hiba adott toleranciával

$$\frac{||x^{k+1} - x^k||}{||x^k||} < TOL$$

◇ Reziduális hiba toleranciával

$$||b - Ax^k|| < TOL$$

A fenti klasszikus leállási feltételeket a teljesség igénye nélkül adtuk meg. Tipikusan ezek kombinációi vannak beépítve adott leállási feltételekként. Mi a továbbiakban a *maximális iteráció szám* mellett a *relatív hiba adott toleranciával* feltételekkel dolgozunk majd. Most nézzük meg a következő példa segítségével, hogy hogyan is működnek ezek.

2.2.2. Példa. Számos elliptikus parciális differenciálegyenlet alkalmas véges differenciás vagy végeselemes diszkretizációs eljárás során a LAER A mátrixa szimmetrikus pozitív definit mátrix. Ezek közül az egyik legfontosabb egyenlet az ún. *Poisson-egyenlet*. Most

ennek egydimenziós verzióját tekintjük, ahol a két végpontban előírt peremérték azonosan nulla, azaz ún. homogén Dirichlet peremfeltétellel dolgozunk. Ennek alakja

$$\begin{aligned} u''(x) + f(x) &= 0 \quad x \in (0,1) \\ u(0) &= 0 \\ u(1) &= 0. \end{aligned}$$

Az egyszerűség kedvéért dolgozzunk az $f(x) = 1$ függvénnyel. A $-u''(x)$ második deriváltat ekvidisztáns rácshálón a standard másodrendű véges differencia hányadossal közelítve, a peremet beépítve, majd a LAER méretét redukálva az alábbi

$$(n+1)^2 \begin{pmatrix} 2 & -1 & 0 & \dots & 0 \\ -1 & 2 & -1 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & -1 & 2 & -1 \\ 0 & \dots & 0 & -1 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \\ 1 \end{pmatrix}$$

redukált LAER-hez jutunk, ahol tömör jelöléssel az $A = \text{tridiag(-1,2,-1)} \in \mathbb{R}^{n \times n}$ szimmetrikus pozitív definit mátrix (további részletekért lásd [1], 10.5.fejezet). A fenti LAER megoldására általános Jacobi, Gauss-Seidel és SOR iterációs megoldó programokat írtunk *maximális iteráció szám* és *relatív hiba adott tolerancia* leállási feltételekkel. Fontos megjegyeznünk, hogy a relatív hibát maximum normában mérjük, de a mérés történhetne más vektornormában is. Lentebb álljon a SOR iteráció megvalósításának kódja.

```
1  %SOR
2  function [it,x]=SOR(A,b,x_0,maxit,tol,omega)
3      it=1;
4      L=tril(A,-1);
5      U=triu(A,1);
6      D=diag(diag(A));
7      err=inf;
8      x=x_0;
9      while err>tol && it<maxit
10         x_uj=(D+omega*L)\(-(omega*U+(omega-1)*D)*x+omega
               *b);
11         err=norm(x_uj-x,inf)/norm(x,inf);
```

```
12            x=x_uj;
13            it=it+1;
14        end
15    end
```

Az általános SOR iterációs programunkban specifikálnunk kell az

(A, b, x0, omega, TOL, maxit)

paramétereket. A fenti LAER esetén A és b alakja adott n-re megkonstruálható.

```
n=2^8;
e=ones(n,1);
A=(n+1)^2*spdiags([-e 2*e -e],[-1 0 1],n,n);
b=e;
```

Az $x0$ kezdeti vektor eddigi tételeink értelmében tetszőleges lehet. Ez ugyanakkor a gyakorlatban mégsem igaz. Ugyanis relatív hibával maximum normában számolva a nevező értéke nulla lenne! Ezért legyen ennek utolsó eleme nullától különböző, például nagyon kicsi.

```
x0=zeros(length(b),1);
x_0(end)=1e-10;
```

További szabad paraméterek a TOL és $maxit$ értékek. A hatékony összehasonlítás érdekében állítsunk be egy kellően nagy maximális iterációszám értéket.

```
maxit=100000;
```

Ekkor a szabad $\omega \in (0, 2)$ relaxációs paraméterérték választásai mellett vizsgáljuk meg, hogy az adott iteráció hány lépés alatt áll le (ha egyáltalán leáll 100000 lépés alatt) adott tolerancia szintek mellett. Azaz összefoglalva a LAER A mátrixa és b jobboldali vektora mellett fixáltuk az egyenletrendszer

- méretét: 256×256,
- kezdeti vektorát $x0 = (0, \ldots, 0, 10^{-10})$,
- maximális iterációszámát $maxit = 100000$.

Adott $TOL = 10^{-1}, 10^{-2}, \ldots, 10^{-6}$ tolerancia szintek mellett különböző ω értékre vonatkozóan iterációs számokra és másodpercben kifejezett futási időre vagyunk kíváncsiak a hatékony összehasonlítás érdekében. Mivel $\omega \in (0, 2)$ esetén erre a feladatra a SOR iteráció konvergens, ezért teszteljünk

$$\omega = \{0.25,\ 0.5,\ 0.75,\ 1,\ 1.25,\ 1.5,\ 1.75\}$$

értékekkel. Természetes kérdés lehet, hogy van-e optimális ω_{opt} érték erre a feladatra vonatkozóan. A SOR iterációra általánosan Poisson-feladatra Dirichlet peremfeltétellel az

$$\omega_{opt} = \frac{2}{1 + \sin\left(\frac{\pi}{n+1}\right)}$$

az optimális ω érték. A feladatra futtatott szimulációink eredményei egységesen a 2.1 és 2.2 Ábrákon láthatóak. Jól látható módon nagy különbségeket a nem optimális ω paraméterek választása esetén nem tapasztalhatunk sem iterációszám tekintetében sem futási idő tekintetében. Az tapasztalhatjuk, hogy az egyes ω értékek növelésével a SOR iteráció egyre jobban működik. A példánkban ω_{opt} értéke

$$\omega_{opt} = \frac{2}{1 + \sin\left(\frac{\pi}{2^8+1}\right)} \approx 1.9758476503.$$

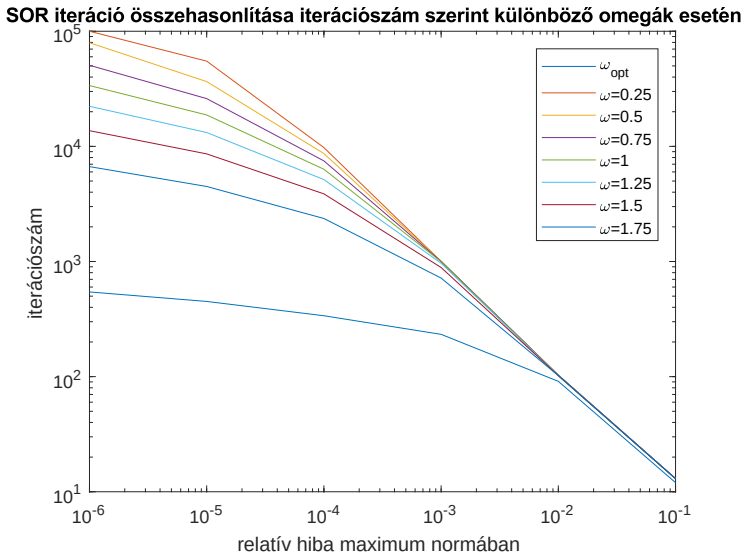
Jól látható módon ha a feladathoz tartozó optimális ω_{opt} érték ismert, akkor mind iterációszám mind futási idő tekintetében nagyságrendbeli javulás várható.

Most érdemes összehasonlítani az optimális értékű SOR iterációt a korábban tanult Jacobi és Gauss-Seidel iterációkkal a fentebb részletezett elvek alapján. A futtatások végeredményeit a 2.3 és 2.4 Ábrák foglalják össze. Az ábrák alapján nyilván való, hogy az optimális paraméterű SOR iteráció mind a Jacobi mind a Gauss-Seidel (speciális SOR) iterációknál kevesebbet iterál ugyanazon tolerancia szint eléréséhez és kevesebb idő alatt is fut le az iteráció. ♣

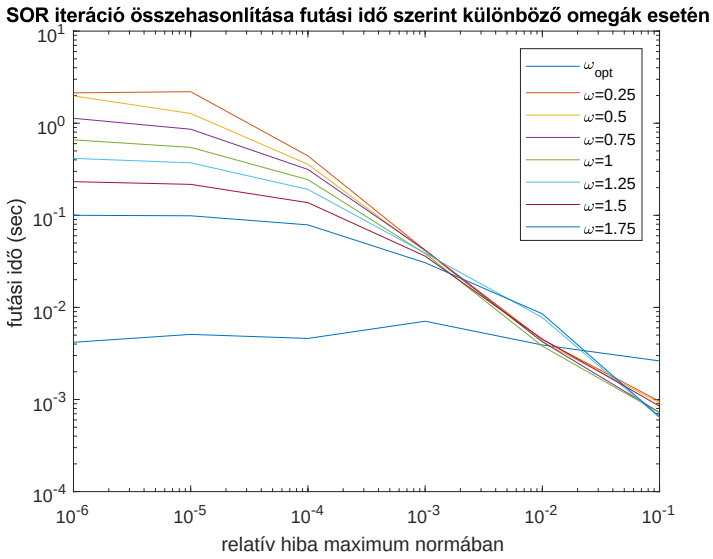
Ahogy a bevezetőben is olvashattuk szimmetrikus pozitív definit együtthatómátrixú LAER-ek számos alkalmazási területen előfordulnak. Egy érdekes alkalmazás az előző 2.2.2. Példa kétdimenziós természetes általánosítása. Ekkor a Poisson-egyenlet segítségével vékony rudak csavarását modellezzük.

2.2.3. Példa. A vékony rudak csavarásának modellezésére szolgál az alábbi

$$\begin{aligned} &\Delta\Phi = -1, \quad (x_1, x_2) \in \Omega, \\ &\Phi|_\Gamma = 0 \end{aligned}$$

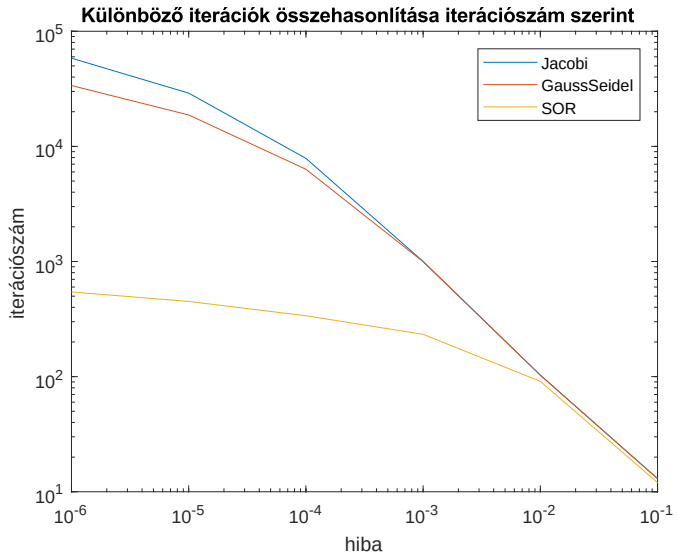


2.1. ábra. Különböző ω értékek esetén az egyes iterációszámok összevetése adott reletív hiba ellenében.

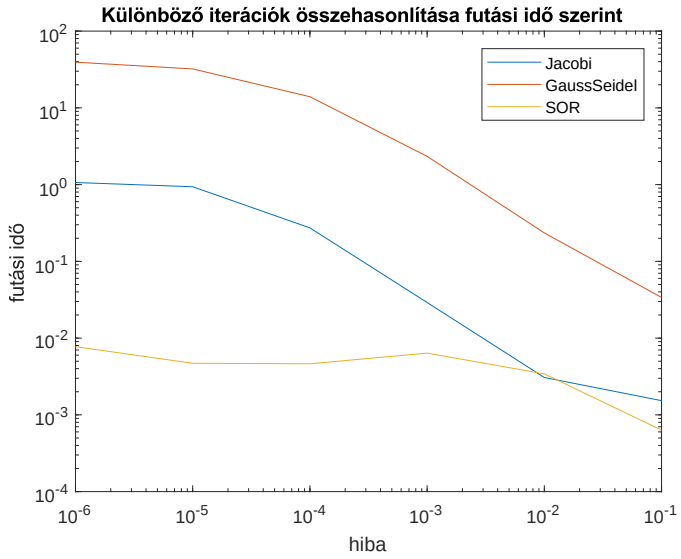


2.2. ábra. Különböző ω értékek esetén az egyes futási idők összevetése adott reletív hiba ellenében.

Poisson-egyenlet. Itt Ω a rúd metszete, míg Γ a tartomány pereme. A Φ segédfüggvényből az eltolódások vektorát kapjuk meg, feltéve ha az egységhosszra vonatkozott csavarási szög a rúd hosszának irányában állandó. Ebben a példában is a 2.2.2. Példában szereplő feltételezésekkel élünk leállási feltételek és hozzá tartozó paramétereik tekintetében. A



2.3. ábra. A három klasszikus iteráció iterációszámainak összevetése adott reletív hiba ellenében.



2.4. ábra. A három klasszikus iteráció egyes futási időjeinek összevetése adott reletív hiba ellenében.

csavarási modellre alkalmazott egység négyzeten mindkét irányban ekvidisztáns rácshálót használó véges differenciás diszkretizációjának technikai részleteit nem ismertetjük, azok a szakirodalomban elérhetőek [[8], 15.3.1. fejezet vagy [4] 2.2. fejezet]. Ekkor az előző

2.2.2. Példához hasonlóan egy LAER-hez jutunk. Ennek alakja

$$(n+1)^2 \begin{pmatrix} B & -I & 0 & \dots & 0 \\ -I & B & -I & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & -I & B & -I \\ 0 & \dots & 0 & -I & B \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n^2-1} \\ x_{n^2} \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \\ 1 \end{pmatrix},$$

ahol

$$B = \begin{pmatrix} 4 & -1 & 0 & \dots & 0 \\ -1 & 4 & -1 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & -1 & 4 & -1 \\ 0 & \dots & 0 & -1 & 4 \end{pmatrix}.$$

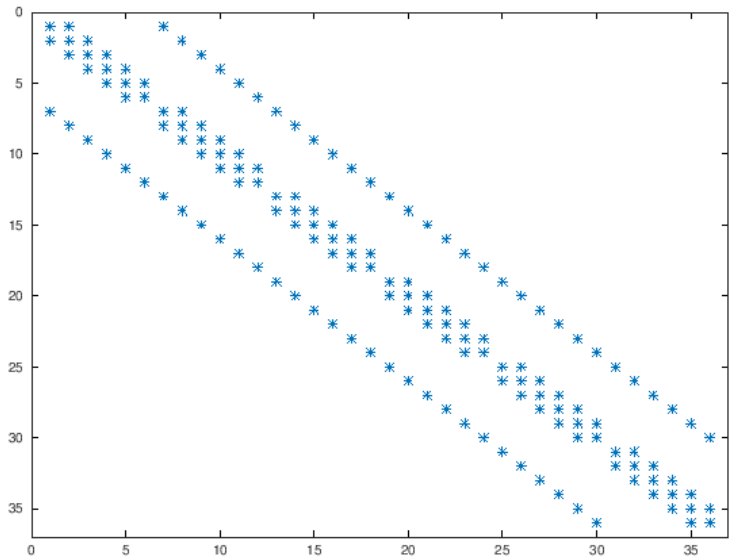
Ekkor a LAER A együtthatómátrixa $n^2 \times n^2$-es méretű, míg a B és I mátrixok $n \times n$-es méretűek. A fenti LAER A együtthatómatrixának megvalósítása az alábbi:

```
e=ones(n,1);
B=spdiags([-e 4*e -e], [-1 0 1], n,n);
C=spdiags([-e -e],[-1 1],n,n);
I=speye(n);
A=(n+1)^2*(kron(I,B)+kron(C,I));
```

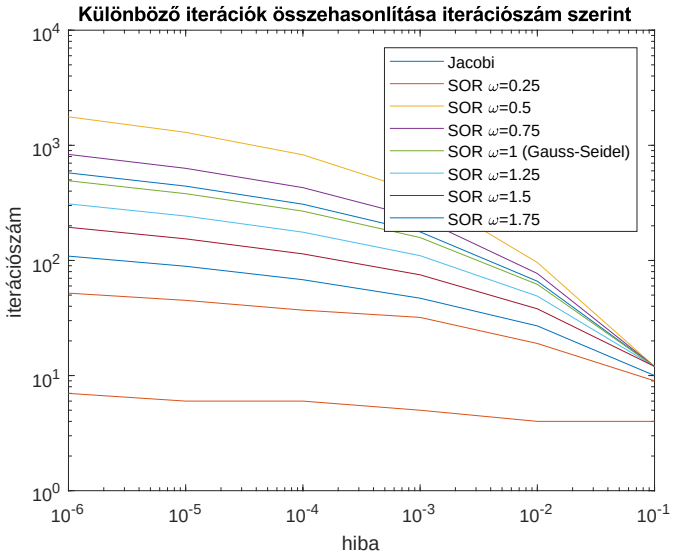
A 2.2.2. Példához hasonlóan ismételten az iterációszámokat és a futási időket vetjük össze az egyes módszerek esetében, annak érdekében, hogy reális következtetéseket vonhassunk le. Ezek eredményeit a 2.6. és 2.7. Ábrákon láthatjuk. Az ábrák alapján a kétdimenziós feladatra azt tapasztaljuk, hogy egy a feladatra jellemző SOR iteráció ω paraméter megválasztása esetén kapjuk adott relatív hiba mellett leggyorsabban leálló iterálást. ♣

Az előző két példa a második deriváltat közelítette másodrendben véges differenciahányados segítségével. A származtatott A mátrixok szimmetrikus, pozitív definit mátrixok voltak további speciális tulajdonságokkal:

- ◇ A ritka mátrix, azaz kevés nemnulla eleme van a mátrix teljes mátrixához képest,
- ◇ A sávszélessége kicsi, azaz a főátlóhoz képest a legtávolabbi mellékátló is közel van (ez a derivált jó lokális közelítésére reflektál).



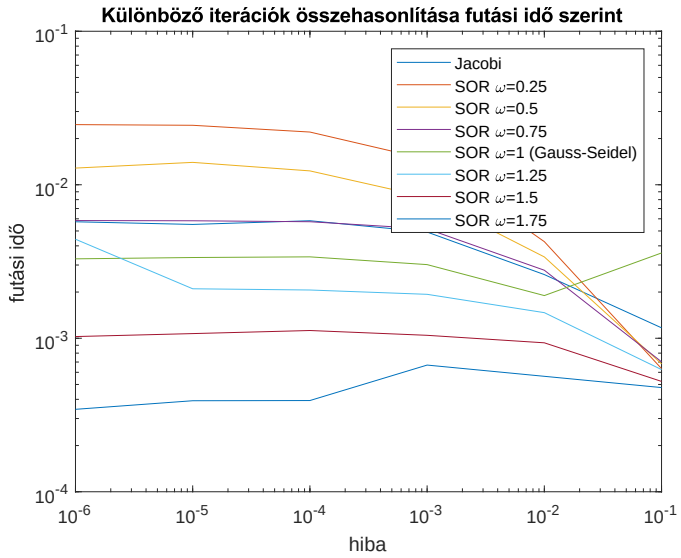
2.5. ábra. A példa együtthatómátrixának nemnullai elemei $n^2 = 36$ esetben.



2.6. ábra. A három klasszikus iteráció iterációszámainak összevetése adott reletív hiba ellenében.

A következő példában a ritkasági és sávszélességi feltételek nemhogy nem fognak teljesülni, de egyenesen teli/sűrű mátrixot vizsgálunk. A példa egy oxfordi numerikus módszerekkel foglalkozó 2015-ös doktori kurzuson hangzott el, melynek videóanyagai szabadon elérhetőek [11].

2.2.4. Példa. A példában a $(0, 1)$ egyenletes eloszlású $0.1,\ 0.5,\ 0.9$ telítettségű (sűrűségű)

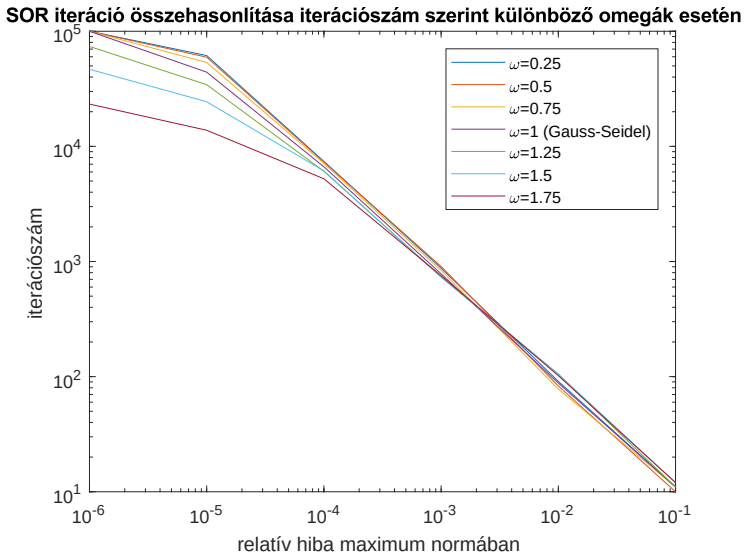


2.7. ábra. A három klasszikus iteráció egyes futási időjeinek összevetése adott reletív hiba ellenében.

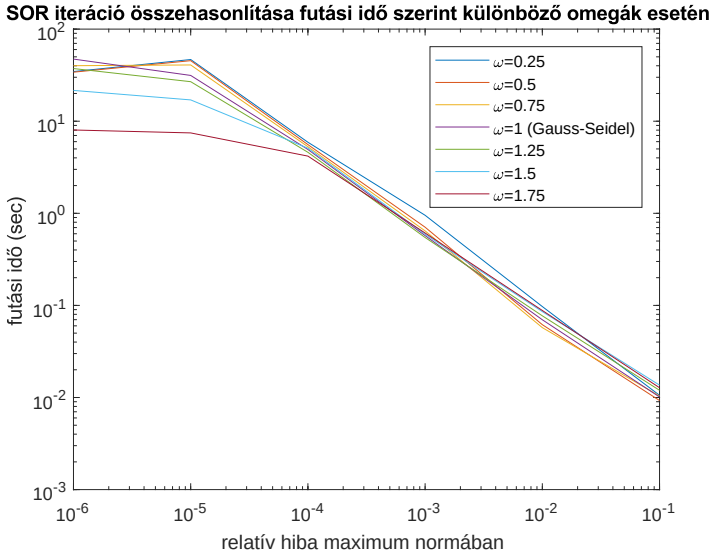
mátrixból készítünk új alkalmas mátrixot. Ennek MATLAB kódja 0.9 telítettség esetén az alábbi:

```
n=2^8;
A = sprandsym(n,0.9);
A=A*A';
```

Fontos megjegyeznünk, hogy itt már csak a SOR iterációval (speciális eseteként a Gauss-Seidel iterációval) foglalkozhatunk, hiszen a Jacobi iteráció esetén 2.1.4. Tétel feltétele nem teljesül szükségképpen. Mind a három telítettségi érték esetén hasonló ábrákat kapunk. Ezért csak a 0.5-ös telítettségi érték eredményeit adjuk meg a 2.8. és 2.9. Ábrákon.



2.8. ábra. Az adott ω paraméterű SOR iteráció iterációszámainak összevetése adott reletív hiba ellenében.



2.9. ábra. Az adott ω paraméterű SOR iteráció egyes futási időjeinek összevetése adott reletív hiba ellenében.

Jól láthatjuk, hogy különböző ω értékek esetén is mind futási időre, mind iterációszámra az egyes SOR iterációk igen hasonló eredményeket szolgáltatnak. ♣

3. fejezet

A konjugált gradiens módszer

Az előző fejezetben klasszikus egylépéses stacionárius iterációk legfontosabbjaival, a Jacobi, Gauss-Seidel és SOR iterációkkal ismerkedhettünk meg. Ebben a fejezetben a célunk egy másfajta gondolat és konstrukció bevezetésével egy hatékonyabb iteratív eljárás szerkesztése. Ez a Krylov-altér iterációs eljárás lesz az ún. *konjugált gradiens módszer*. Mielőtt a módszer alapgondolatát, tulajdonságait és hatékonyságát ismernénk meg tételeken és példákon keresztül álljon itt annak igen érdekes története.

A már korábban említett és hivatkozott doktori numerikus kurzuson [11] ecsetelte az előadó (a numerikus lineáris algebra világhírű képviselője Trefethen professzor úr, aki egyben az első MATLAB tulajdonos is volt), hogy a módszert leíró eredeti 1952-es Hestenes és Stiefel cikket [3] a szakma vegyes fogadtatásban részesítette. Az algoritmust magát ismerték, de nem tulajdonítottak nagy jelentőséget neki és más klasszikus módszereket használtak helyette (például a direkt Gauss-eliminációs eljárást). Bő 20 évvel később John Reidnek köszönhetően ismerték fel az iteratív módszer jelentőségét, ugyanis ahogy az 1. Fejezet 1.1. Táblázatában is láthattuk a LAER-ek méretének növekedésével nagy n-ek esetén tapasztalni kezdték a módszer hatékonyságát. Habár a módszer komplexitása általános esetben nem változik, de az esetek többségénél $\mathcal{O}(n^2)$ műveletigényt tapasztaltak. A jelenség okát később meg is magyarázzuk. A módszerről további történelmi érdekességeket Golub és O'Leary cikkében találhatunk [2].

Most térjünk rá a konjugált gradiens módszer alapgondolatára. Ehhez vezessük be a már említett *Krylov-alteret*.

3.0.1. Definíció. *A $b \in \mathbb{R}^n$ vektor által generált k-adik Krylov-altér*

$$\mathcal{K}_k = \langle b, Ab, \ldots, A^{k-1}b \rangle = span\{b, Ab, \ldots, A^{k-1}b\},$$

ahol $A \in \mathbb{R}^{n \times n}$.

Ismeretes, hogy szimmetrikus pozitív definit mátrixok sajátértékei is pozitívak. Ez pedig az alábbi definíciót motiválja.

3.0.2. Definíció. *Legyen* $A \in \mathbb{R}^{n \times n}$ *szimmetrikus pozitív definit mátrix. Ekkor az*

$$||x||_A = \sqrt{x^T A x}$$

kifejezést A*-normának nevezzük.*

A fenti 3.0.2. Definícióban szereplő $|| \cdot ||_A$ kifejezés normát definiál $\mathbb{R}^n$-en. A fenti két definíció akkor nyerne igazi értelmet, ha a Krylov-altérből vett numerikus iteráló vektorral számított hibavektor az A-normában minimális lenne, azaz

$$||e^k||_A = ||x^k - x^*||_A$$

norma minimális lenne, ahol $x^k \in \mathcal{K}^k$. Ehhez azonban tekintsük a módszer pszeudokódját, ami egy kívánt $x^k \in \mathcal{K}^k$ vektort állít elő. A kellemesebb forma okán a lenti pszeudokódban az alsó indexek az eddigi felső indexek iterált érték szerepét töltik be.

$x_0 = 0, r_0 = b, p_0 = r_0$

for $k = 1, 2, \ldots,$

$\alpha_k = (r_{k-1}^T r_{k-1})/(p_{k-1}^T A p_{k-1})$	lépéshossz
$x_k = x_{k-1} + \alpha_k p_{k-1}$	közelítő megoldás
$r_k = r_{k-1} - \alpha_k A p_{k-1}$	reziduális
$\beta_k = (r_k^T r_k)/(r_{k-1}^T r_{k-1})$	növekmény
$p_k = r_k + \beta_k p_{k-1}$	keresési irány

end

Programozási szempontból jól látható, hogy a pszeudokód pár soros MATLAB kóddal könnyedén megvalósítható. Nagy előnye, hogy ténylegesen vektorműveletekkel dolgozunk. Teljes, strukturálatlan A mátrix esetén a mátrix-vektor művelet lépésenként körülbelül n^2 FLOPS-ot igényel. Ugyanakkor ha A ritka és strukturált, ez n FLOPS-ra csökkenthet. Utóbbi esetben a teljes eljárás során már az $\mathcal{O}(n^2)$ elérhető. Később említünk tulajdonságot, ahol ezzel a jól strukturáltsággal az $\mathcal{O}(n^2)$ műveletigény elérhető.

Most a korábban említett $|| \cdot ||_A$ minimális tulajdonság irányába teszünk lépéseket. Ehhez kimondunk egy tételt, melyet nem bizonyítunk.

3.0.3. Tétel. *Alkalmazzuk a konjugált gradiens módszert a (2.1) LAER-re, ahol A szimmetrikus pozitív definit mátrix. Mindaddig míg az iteráció nem konvergált (azaz $r^{k-1} \neq 0$), az algoritmus végrehajtódik és az alábbi tulajdonságok érvényesek:*

$$\begin{aligned}\mathcal{K}_k &= \langle x^1, x^2, \ldots, x^k \rangle = \langle p^0, p^1, \ldots, p^{k-1} \rangle \\ &= \langle r^0, r^1, \ldots, r^{k-1} \rangle = \langle b, Ab, \ldots, A^{k-1}b \rangle.\end{aligned}$$

Továbbá a reziduálisak ortogonálisak, azaz

$$(r^k)^T r^j = 0 \quad j < n$$

és a keresési irányok ún. A-konjugáltak, azaz

$$(p^k)^T A p^j = 0 \quad j < n.$$

Bizonyítás. A bizonyítás a [10] könyv Theorem 38.1. tétel kimondása után található. □

Már nincs más hátra, minthogy bizonyítsuk, hogy a konjugált gradiens módszer minimalizálja $||e||_A$-t.

3.0.4. Tétel. *Alkalmazzuk a konjugált gradiens módszert a (2.1) LAER-re, ahol A szimmetrikus pozitív definit mátrix. Ha az iteráció még nem konvergált, akkor x^k az egyértelmű elem $\mathcal{K}^k$-ből mely minimalizálja $||e^k||_A$-t. A konvergencia monoton, azaz*

$$||e^k||_A \leq ||e^{k-1}||_A \tag{3.1}$$

és $e^k = 0$ felvétetődik valamely $k \leq n$ esetén.

Bizonyítás. A 3.0.3. Tétel alapján $x^k \in \mathcal{K}^k$. Ahhoz hogy megmutassuk, hogy x^k az egyértelmű elem $\mathcal{K}^k$-ből mely minimalizálja $||e^k||_A$-t tekintsünk egy tetszőleges

$$\tilde{x} = x^k + \Delta x \in \mathcal{K}^k$$

elemet. Ennek hibavektorára az

$$e = x - x^* = x^k x^* + \Delta x = e^k + \Delta x$$

összefüggés adódik. Ekkor

$$\begin{aligned}||e||_A^2 &= \left(e^k + \Delta x\right)^T A \left(e^k + \Delta x\right) \\ &= (e^k)^T A e^k + (\Delta x)^T A (\Delta x) + 2(e^k)^T A (\Delta x).\end{aligned}$$

A fenti kifejezés utolsó tagjában a 3.0.3. Tétel alapján $2(e^k)^T A(\Delta x) = 2(r^k)^T(\Delta x)$. Ez pedig r^k skalárszorzata egy $\mathcal{K}^k$-beli elemmel és a hivatkozott tételben szereplő ortogonalitási tulajdonság miatt ez nulla, így a fenti összefüggés utolsó tagja elhagyható. Azaz

$$||e||_A^2 = (e^k)^T A e^k + (\Delta x)^T A(\Delta x).$$

A fenti összefüggés második tagja függ csak Δx-től és mivel A pozitív definit, ezért ennek értéke nemnegatív. Egyenlőség csak a $\Delta x = 0$ esetben áll fenn, azaz ha $x = x^k$. Így az $||e||_A$ norma pontosan akkor minimális, ha $x = x^k$, azaz az egyértelműségi részt beláttuk.

A (3.1) monotonitási tulajdonság a $\mathcal{K}^k \subset \mathcal{K}^{k+1}$ tartalmazás következménye és $\mathcal{K}^k$ a k dimenziós $\mathbb{R}^n$ részhalmaza, ezért a konvergencia legföljebb n lépés alatt elérhető. □

A fenti 3.0.4. Tétel segítségével éppen most láttuk be, hogy a konjugált gradiens módszernek van bizonyos optimalitási tulajdonsága. Mégpedig az $||e^k||_A$ normáját minimalizálja k lépéssel az összes $x \in \mathcal{K}^k$ vektorra nézve. A korábban feltüntetett pszeudokódban szerepeltek olyan kifejezések minthogy "lépéshosszúság" és "keresési irány". Az iteráció ugyanis felfogható egyfajta olyan standard alakú algoritmusként, mely egy nemlineáris függvényt minimalizál. Az iterációs eljárás lelkét az

$$x^k = x^{k-1} + \alpha^k p^{k-1}$$

sor adja. Ez egy tipikusan előforduló egyenlet optimalizálásban, amelyben az aktuális x^{k-1} közelítést updateljük α^k hosszal (lépéshossz) p^{k-1} irányban (keresési irány). Ilyen lépések ismétlésével a konjugált gradiens módszer próbálja a nemlineáris függvény minimumát megtalálni. De melyik függvény minimumát?

A 3.0.4. Tétel eredménye alapján az $||e||_A$-ra vagy ekvivalens módon az $||e||_A^2$-re gondolhatnánk. Habár $||e||_A^2$ x függvénye, de x^* ismerete nélkül nem tudjuk kiértékelni. Márpedig a konjugált gradiens módszer nem lenne jó eljárás egy függvény optimalizálására, ha nem lehetne kiértékelni. Az bizonyos, hogy adott A, b és x esetén a

$$\varphi(x) = \frac{1}{2} x^T A x - x^T b \tag{3.2}$$

mennyiség kiértékelhető. Most számítsuk ki $||e^k||_A^2$ érékét.

$$\begin{aligned} ||e^k||_A^2 = (e^k)^T A e^k &= \left((x^k)^T - x^*\right) A (x^k - x^*) = \left(x^* - (x^k)^T\right) A (x^* - x^k) \\ &= (x^k)^T A x^k - 2(x^k)^T A x^* + (x^*)^T A x^* \\ &= (x^k)^T A x^k - 2(x^k)^T b + (x^*)^T b = 2\varphi(x^k) + (x^*)^T b. \end{aligned}$$

Azaz így a $\varphi(x)$ az $||e||_A^2$ normától egy kettes szorzóban és egy plusz $(x^*)^T b$ tagban különbözik. De $||e||_A^2$-hez hasonlóan a minimumot egyértelműen elérhetjük. Ebben az esetben konkrétan $x = -(x^*)^T b/2$ esetén.

Azaz a konjugált gradiens módszer egy olyan iteratív eljárásként értelmezhető, amely minimalizálja a $\varphi(x)$ kvadratikus függvényt. Minden egyes iterációs lépés során egy

$$x^k = x^{k-1} + \alpha^k p^{k-1}$$

iteráltat számítunk ki, mely minimalizálja $\varphi(x)$-et minden x-re az $x^{k-1} + \langle p^{k-1} \rangle$ egydimenziós térben. Az

$$\alpha^k = \left((r^{k-1})^T r^{k-1}\right) / \left((p^{k-1})^T A p^{k-1}\right)$$

formula az α^k értékének optimalitását biztosítja az összes α lépéshossz esetén. A konjugált gradiens módszert a p^{k-1} keresési irány teszi igazán kiemelkedő algoritmussá, hiszen ez az irány a $\varphi(x)$-et nemcsak az egydimenziós $x^{k-1} + \langle p^{k-1} \rangle$ térben minimalizálja, hanem a teljes $\mathcal{K}^k$ Krylov-altérben.

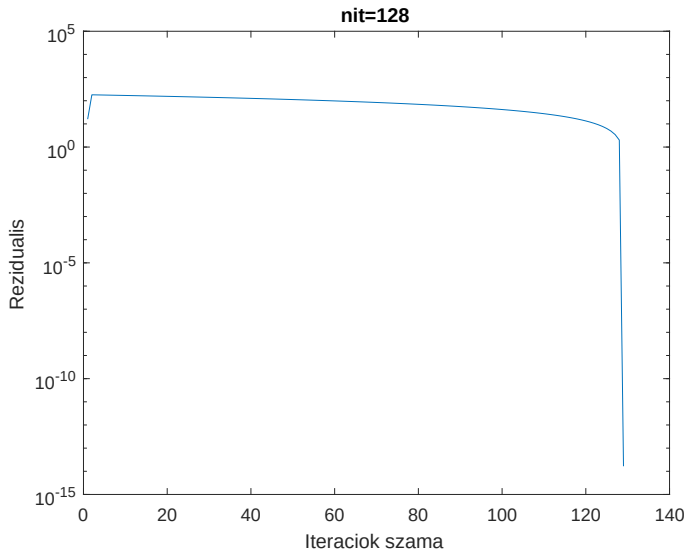
3.1. Numerikus példák

Az előző alfejezetbeli 3.0.4. Tétel értelmében várhatjuk, hogy $||\cdot||_A$ normában a módszer konvergáljon szimmetrikus pozitív definit együtthatómátrix esetén. Fontos megjegyeznünk, hogy a módszer pszeudokódjában a kezdeti vektor az azonosan nulla vektor volt, de a módszer konvergál tetszőleges kezdeti vektor mellett is. Az eddigi relatív hiba adott tolerancia leállási feltételt a reziduális hiba tolerancia feltételre cseréljük, és így az azonosan nulla kezdeti vektor nem okoz gondot.

A 2. Fejezet 2.2.2. és 2.2.4. Példáit vesszük górcső alá több módosítással.

- ◇ Azonosan nulla kezdeti vektorral indul az iteráció.
- ◇ Az egyik leállási feltétel a reziduális hiba adott toleranciával.

3.1.1. Példa. A 2.2.2. Példánál láthattuk, hogy a klasszikus iterációs eljárások közül mind futási idő, mind iterációszám tekintetében az ω_{opt} SOR iteráció volt a leghatékonyabb. A fenti módosításokkal az új futtatások után ismételten ez az állítás marad érvényben. A releváns információhoz rögzítettük, hogy a tolerancia szint legyen 10^{-6}. Ekkor ω_{opt} SOR iteráció 870 alkalommal iterált ebben az esetben. Vessük össze, hogy a konjugált



3.1. ábra. Az iterációk száma a konjugált gradiens módszer esetén 10^{-6} reziduális hiba mellett.

gradiens módszer hogyan teljesít. A reziduális hibára vonatkozó eredményt a 3.1. Ábra mutatja.

A 3.1. Ábra jól mutatja, hogy a 128. iterálás után csökken a reziduális hiba 10^{-6} tolerancia szint alá. Azaz a konjugált gradiens módszer körülbelül egy hetednyi iterálást kíván az optimális paraméterű SOR iterációval szemben. ♣

Következzen most a másik példa.

3.1.2. Példa. Az új feltételekkel futtatva a 2.2.4. Példát a legkisebb futási idejű ω paraméterű sor iterációra az alábbi eredmények adódnak:

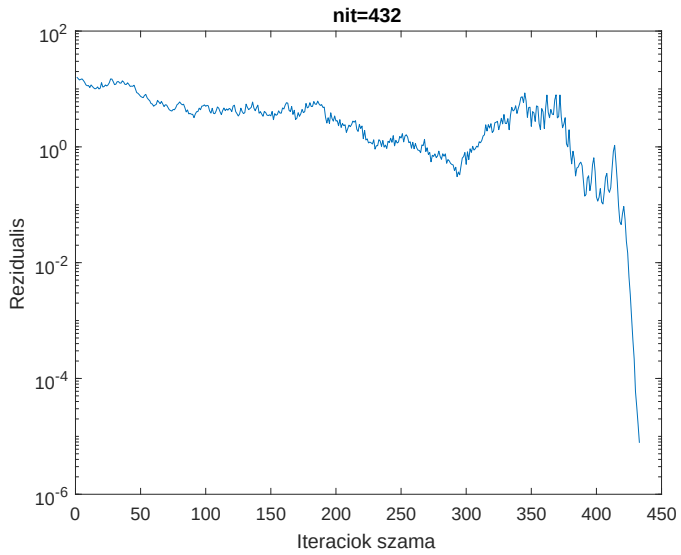
```
Siteracio =

      48038.00      65737.00      83436.00     100000.00
      100000.00     100000.00


Stoc =

         53.74         19.58         19.49         18.13
          0.00          0.00
```

Azaz erre a feladatra a SOR iteráció 48038 lépést tesz meg a 10^{-1} tolerancia szint eléréséhez és ezt 53.74 másodperc alatt teszi meg. Az utolsó még a maxit 100000-es maximális iteráció utoljára 10^{-3} tolerancia szint mellett futott le 83436 iterációval 92.81 másodperc alatt. Azaz a 10^{-6}-os tolerancia leállási feltételhez nagyságrendekkel nagyobb maximális iterációt kell beállítani. Ezt nem kíséreltük meg. Helyette a feladatra futtattuk a konjugált gradiens módszert.



3.2. ábra. Az iterációk száma a konjugált gradiens módszer esetén 10^{-6} reziduális hiba mellett.

A 3.2. Ábra jól mutatja, hogy a konjugált gradiens módszer 432 iterálást kíván és mindössze 0.38 másodperc alatt futott le az algoritmus. ♣

A fenti két példa kiválóan érzékelteti, hogy a konjugált gradiens módszer jóval hatékonyabb módszer a klasszikus iterációkkal szemben. A következő alfejezetben a 3. Fejezet elején említett átlagos $\mathcal{O}(n^2)$ műveletigény magját értjük meg.

3.2. Specifikus tételek

A 3.0.1. Definícióból látható, hogy egy $\tilde{x} \in \mathcal{K}^k$ előáll $\tilde{x} = p(A)b$ alakban, ahol $p(A)$ egy legfeljebb $k-1$-edfokú polinom. Ekkor a konjugált gradiens módszer közelítő tulajdonsága az alábbi kérdésre fogalmazható át: az $Ae^0 = b$ tulajdonság figyelembevételével keresendő

egy $p^k \in \mathcal{P}_k$, $p^k(0) = 1$ tulajdonságú polinom a legfeljebb k-adfokú polinomok osztályából, hogy

$$||p^k(A)e_0||_A \tag{3.3}$$

minimális legyen. A korábban bizonyított 3.0.4. Tételből egy másik konvergencia tételt is igazolhatunk. Ezt a tételt csupán kimondjuk.

3.2.1. Tétel. *Ha a konjugált gradiens módszer még nem konvergált a k-adik lépés előtt, akkor a (3.3) problémának létezik egyértelmű $p^k \in \mathcal{P}_k$ megoldása és az x^k iterált az $e^k = p^k(A)e^0$ hibaegyenlet igaz ugyanezen polinommal. Továbbá*

$$\frac{||e^k||_A}{||e^0||_A} = \inf_{p \in P_k} \frac{||p(A)e^0||_A}{||e^0||_A} \leq \inf_{p \in P_k} \max_{\lambda \in \sigma(A)} |p(\lambda)|$$

ahol $\sigma(A)$ jelöli az A spektrumát.

Bizonyítás. A bizonyítás a [10] könyv Theorem 38.3. tétel kimondása után található. □

A fenti 3.2.1. Tétel lényegében azt állítja, hogy a konjugált gradiens módszere konvergencia rátájának mértéke nagyban függ A spektrumától, azaz a sajátértékeinek halmazától. A jó spektrum olyan, melyen a $p^k \in \mathcal{P}_k$ polinomok kicsik és még kisebbek lesznek k növelésével. Ez tipikusan két esetben is megtörténhet.

- ◇ A sajátértékek jól klaszterezettek.
- ◇ A sajátértékek kellően szeparáltak az origótól.

Az ezekre vonatkozó állításokat bizonyítások nélkül közöljük.

3.2.2. Tétel. *Ha az $A \in \mathbb{R}^{n \times n}$ szimmetrikus pozitív definit mátrixnak csak k sajátértéke különböző, akkor a konjugált gradiens módszer konvergál legfeljebb k lépés alatt.*

Bizonyítás. A bizonyítás a [10] könyv Theorem 38.4. tétel kimondása után található. □

3.2.3. Tétel. *Alkalmazzuk a konjugált gradiens módszert a (2.1) LAER-re, ahol A szimmetrikus pozitív definit mátrix. Ekkor*

$$\frac{||e^k||_A}{||e^0||_A} \leq 2 / \left[\left(\frac{\sqrt{\kappa}+1}{\sqrt{\kappa}-1} \right)^n + \left(\frac{\sqrt{\kappa}+1}{\sqrt{\kappa}-1} \right)^{-n} \right] \leq 2 \left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1} \right)^n,$$

ahol κ az A mátrix spektrálnormájával számított kondíciószám.

Bizonyítás. A bizonyítás a [10] könyv Theorem 38.5. tétel kimondása után található. □

A fenti 3.2.3. Tétel a szakirodalomból a leghíresebb konvergencia tétel a módszerrel kapcsolatban. Mivel

$$\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1} \approx 1 - \frac{2}{\sqrt{\kappa}}$$

ha $k \to \infty$. Ez azt jelenti, ha κ elgendően nagy, de nem túlságosan nagy, akkor a várt tolerancia szint eléréséhez $\mathcal{O}(\sqrt{\kappa})$ iteráció várható. Ez persze csupán egy felső korlát. Jó jobb oldal esetén (kevésbé gyakori) vagy jól klaszterezett spektrum esetén (gyakori) a konvergencia gyorsabb lehet.

Ebben az alfejezetben a konjugált gradiens módszer konvergencia rátájának növekedési lehetőségét láttuk, mely jelentősen függ a spektrumtól és a kondíciószámtól is. Még hatékonyabban működő konjugált gradiens módszer változatokat tipikusan két nagy technika alkalmazásával érthetünk el, melyek egyenként egy-egy szakdolgozatnyi terjedelmet kívánnának:

- ◇ Párhuzamos implementálás.
- ◇ Prekondícionálás.

Előbbi nagyméretű szuperszámítógépek esetén teszi lehetővé algoritmusunk használatát egyes folyamatokat párhuzamosan, külön magokon futtatva. Utóbbi pedig a nagy κ kondíciószámmal rendelkező LAER kondíciószámát csökkenti. Ezekről a technikákról bővebben a [7] könyv 245–382. oldalain olvashatunk.

4. fejezet

Összefoglalás

A szakdolgozat speciális együtthatómátrixú (szimmetrikus pozitív definit) LAER-ek iteratív megoldási módszereivel foglalkozott.

A 2. Fejezetben klasszikus iteratív módszereket konstruáltunk és azokhoz kapcsolódó konvergenciatételeket bizonyítottunk. Különböző példák megoldása során az implementálási kérdések fontosságát is érzékeltettük. A módszereket több kritériumot is figyelembe véve hasonlítottuk össze.

A 3. Fejezetben a mai napig a leggyakrabban használt Krylov-altér iterációs eljárással foglalkoztunk, a konjugált gradiens módszerrel. Az elméleti felépítésen és konvergenciához kapcsolódó tételek ismertetésén túl a módszer tényleges erejét példákon keresztül is demonstráltunk a klasszikus iterációkkal szemben.

Irodalomjegyzék

[1] Faragó I., Horváth R., Numerikus módszerek, Typotex, 2013

[2] G. H. Golub and D. P. O'Leary, Some history of the conjugate gradient and Lanczos methods, SIAM Review 31 (1989), 50–100.

[3] M. R. Hestenes and E. Stiefel, Methods of conjugate gradients for solving linear systems, J. Res. Nat. Bur. Stand. 49 (1952), 409–436.

[4] Karátson J., Horváth R., Izsák F., Parciális differenciálegyenletek numerikus módszerei számítógépes alkalmazásokkal, Typotex, 2013

[5] A. M. Ostrowski, On the linear iteration procedures for symmetric matrices, Rend. Mat. e Appl. (5)14 (1954), 140–163.

[6] E. Reich, On the convergence of the classical iterative method for solving linear simultaneous equations, Ann. Math. Statist., 20 (1949), 448–451.

[7] Y. Saad, Iterative Methods for Sparse Linear Systems, Second Edition, SIAM, 2003

[8] Stoyan G, Takó G., Numerikus módszerek III., Typotex, 1997

[9] TOP500 Supercomputer Sites, 2018. november

[10] N. L. Trefethen, D. Bau, Numerical Linear Algebra, SIAM, 1997

[11] N. L. Trefethen, Scientific Computing for DPhil Students kurzus (http://podcasts.ox.ac.uk/series/scientific-computing-dphil-students), Oxfordi Egyetem, 2015