

EÖTVÖS LORÁND TUDOMÁNYEGYETEM

TERMÉSZETTUDOMÁNYI KAR

ALAPVETŐ ITERÁCIÓS ELJÁRÁSOK LINEÁRIS EGYENLETRENDSZEREK MEGOLDÁSÁRA

SZAKDOLGOZAT

Ruzsics László

Matematika B.Sc., elemző szakirány

Témavezető: **Kurics Tamás**, adjunktus

Alkalmazott Analízis és Számításmatematikai Tanszék



Budapest

2012

Tartalomjegyzék

| | |
|--|-----------|
| 1. Lineáris egyenletrendszerek | 2 |
| 2. Megoldási módszerek | 4 |
| 2.1. Direkt módszerek | 4 |
| 2.1.1. Gauss-elimináció | 4 |
| 2.1.2. LU-felbontás | 5 |
| 2.2. Iterációs módszerek | 6 |
| 3. Alapvető iterációs módszerek: Jacobi-és Gauss-Seidel-iterációk | 10 |
| 3.1. Jacobi módszer | 11 |
| 3.2. Gauss-Seidel-iteráció | 11 |
| 4. Relaxációs módszerek | 13 |
| 4.1. JOR módszer | 13 |
| 4.2. SOR módszer | 14 |
| 4.3. Relaxációs módszerek tridiagonális mátrixokra | 16 |
| 5. Az eddigi módszerek kipróbálása különböző mátrixokon | 21 |
| 5.1. Tridiagonális mátrix | 21 |
| 5.2. Iterációk közti különbségek | 26 |
| 5.3. Iterációk összehasonlítása egy eddigieknél nagyobb méretű mátrix esetén | 27 |
| 6. Összegzés | 29 |
| 7. Köszönetnyilvánítás | 30 |

1. fejezet

Lineáris egyenletrendszerek

Lineáris egyenletrendszerek nagyon sok helyről eredhetnek. A [3] könyv a következő témakörökkel kapcsolatba említi: gyártás, mechanika, ökológia, gazdaság és villamosságtan. Egy egyenletrendszer általános alakja $Ax = f$. Részletesen:

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \dots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{bmatrix},$$

ahol az A mátrix egy $n \times n$ -es mátrix. Az x egy $n \times 1$ -es oszlop mátrix, ami az ismeretleneket tartalmazza. Az f mátrix szintén egy $n \times 1$ -es oszlop mátrix. Az A és az f mátrix elemei lehetnek akár valósak, de akár komplexek is. Egyenletrendszerek származhatnak például differenciálegyenletekből is:

$$-u''(x) = f(x) \quad x \in (0, 1)$$

$$u(0) = \alpha$$

$$u(1) = \beta.$$

A $(0, 1)$ intervallumot felosztjuk N egyenlő részre. Jelöljük $x_0 = 0$, $x_1 = h$, $x_2 = 2h$, \dots , $x_N = 1$, ahol $h = \frac{1}{N}$. Így a második derivált közelítése

$$-u''(x_j) \approx \frac{-u(x_{j-1}) + 2u(x_j) - u(x_{j+1}))}{h^2}$$

lesz. A két peremértéket ismerjük. Ebből felírhatunk egy lineáris egyenletrendszert $u(x_j) = u_j$ jelöléssel, ami a következőképpen néz ki:

$$\begin{bmatrix} 2 & -1 & 0 & \dots & 0 \\ -1 & 2 & -1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & -1 & 2 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{bmatrix}.$$

A fent előállt mátrixot tridiagonális mátrixnak hívjuk, mivel a diagonálison és a közvetlenül mellette lévő elemeken kívül nem található nem nulla elem a mátrixban. Ezekkel később részletesebben foglalkozok a relaxációs iterációs módszereknél. Ennek a feladatnak a megoldását a következőképp is megkaphatjuk:

$$x = A^{-1}b.$$

Ami tulajdonképpen egy mátrixinverz-számítást és egy mátrixszorzást jelent. De az inverzmátrix-számítás már kis mátrixokra nézve is nagy műveletigénnyel jár együtt, ezért érdemesebb más módszerhez folyamodni.

2. fejezet

Megoldási módszerek

2.1. Direkt módszerek

Az egyik ilyen megoldási módszer családot direkt módszereknek nevezzük. Jellemzőjük a véges sok lépés, majd az eljárás végén kerekítési hibáktól eltekintve a pontos megoldást kapjuk eredményül. Fő problémájuk a nagy tárigény lehet.

2.1.1. Gauss-elimináció

Az egyik legismertebb direkt módszer a Gauss-elimináció. Lényegében abból áll, hogy olyan megengedett mátrixműveleteket hajtunk végre, amik során egy felső háromszög mátrixot hozunk létre, így az utolsó változóból visszafejtve meg tudjuk mondani az egyenletrendszerünk pontos megoldását. Mivel a szakdolgozat témája az iterációs módszerek, a Gauss-eliminációt csak példa szintjén ismertetem az alábbi mátrixra:

$$\begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}.$$

Ebből az első sor első elemének segítségével elimináljuk a többi sor első elemét:

$$\begin{bmatrix} 2 & -1 & 0 \\ 0 & 3/2 & -1 \\ 0 & -1 & 2 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 3/2 \\ 1 \end{bmatrix}.$$

Most a második sor második elemét tartjuk meg és a harmadik sor második elemét tüntetjük el:

$$\begin{bmatrix} 2 & -1 & 0 \\ 0 & 3/2 & -1 \\ 0 & 0 & 4/3 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 3/2 \\ 2 \end{bmatrix}.$$

Így már visszahelyettesítésekkel a következő megoldást kapjuk, hogy

$$x_3 = 1,5, \quad x_2 = 2, \quad x_1 = 1,5.$$

Hátránya a Gauss-eliminációnak a nagy műveletigény, mert minden egyes lépés alkalmával végig kell mennünk a mátrix sorain újra. Majd végén még egyszer az egészen, hátulról visszafelé helyettesítéssel, hogy megkapjuk a megoldást. Azaz egy $n \times n$ -es mátrix esetén t különböző f érték mellett a Gauss eliminációnak

$$\frac{n^3}{3} + tn^2 - \frac{n}{3}$$

műveletre van szüksége (lásd [1]). Ami egy 10×10 mátrix esetén maximum 1400 műveletet jelenthet.

2.1.2. LU-felbontás

Gauss-elimináció egyik változata az LU-felbontás, amiről a [3] könyvben lehet részletesebben olvasni, és aminek a célja, hogy az A mátrixot egy felső (U) és egy alsó (L) háromszög mátrix szorzatára bontsa fel. Az U mátrixot már eleve ismerjük, hiszen ezt számoltuk ki a Gauss-eliminációval. Ha minden egyes műveletet, amit végrehajtottunk az A mátrix egy során egy oszlopmátrixba foglaljuk, akkor ezekből a vektorokból készített mátrixból megkapjuk az L alsó háromszög mátrixot is. Előző példából az első lépésnél $1/2$ -el szoroztuk meg az első sort és hozzáadtuk a másodikhoz, míg a harmadik sorral nem csináltunk semmit ezért

$$L_1 = \begin{bmatrix} 1 & 0 & 0 \\ 1/2 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

lesz. Míg a második lépésnél $-2/3$ -dal szoroztuk meg a második sort és hozzáadtuk a harmadik sorhoz így az

$$L_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -2/3 & 1 \end{bmatrix}$$

lesz. Így az $L_2L_1A = U$ egyenletből A -t kifejezve kapjuk, hogy

$$A = L_1^{-1}L_2^{-1}U,$$

aminek jobboldala nem más lesz más, mint

$$\begin{bmatrix} 1 & 0 & 0 \\ -1/2 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 2/3 & 1 \end{bmatrix} \begin{bmatrix} 2 & -1 & 0 \\ 0 & 3/2 & -1 \\ 0 & 0 & 4/3 \end{bmatrix}.$$

Ebből L_1^{-1} és L_2^{-1} összeszorozva kapjuk, hogy az L mátrix nem lesz más, mint:

$$L := \begin{bmatrix} 1 & 0 & 0 \\ -1/2 & 1 & 0 \\ 0 & 2/3 & 1 \end{bmatrix}.$$

Ennek a kiszámítása időpocsékolásnak tűnhet, de ha esetleg f értékei valami oknál fogva megváltoznak, akkor már tudjuk az A mátrix felbontását, amivel már könnyedén meg tudjuk mondani az egyenletrendszer megoldását, anélkül hogy még egyszer le kéne futtatnunk a Gauss-eliminációt:

$$LUx = f,$$

amiből, ha $Ux := y$

$$Ly = f,$$

és ezt már könnyen meg tudjuk oldani, mivel L egy alsó háromszög mátrix, amiből az L mátrix első sorától kezdve meg tudjuk kapni y értékeit, és majd ebből:

$$Ux = y,$$

amit szintén könnyedén, egy visszafejtéssel meg tudunk oldani. Így, ha az LU felbontás adott, akkor az egyenletrendszer megoldásának műveletigénye igen alacsony lesz, pontosabban $O(n^2)$.

2.2. Iterációs módszerek

Az iterációs módszerek főbb jellemzői a pontos végeredmény közelítése, akár végtelen sokszor ismételtetű lépésekkel és a kisebb tárigény. Igaz, hogy nem a pontos eredményt

adják a direkt módszerekkel szemben, de erre nem is mindig van szükség. Mért adatok esetében a beérkező adatok is hibával terhelték, így nem biztos, hogy érdemes a pontos megoldást kiszámítani. A stacionárius iterációs módszerek általános alakja:

$$x^{(m+1)} = Bx^{(m)} + b \quad (m = 0, 1, 2, \dots),$$

ahol A egy $n \times n$ -es mátrix és iterációs mátrixnak hívjuk, az $x^{(m+1)}$ és f $n \times 1$ -es oszlopvektorok. A kiinduló vektor az $x^{(0)}$ általában az azonosan nulla oszlopvektor, de ha esetleg más információval rendelkezünk, akkor bármilyen vektort megadhatunk, mint kezdő vektor.

Pár dologra szükség van, hogy az iteráció az egyenletrendszerünk megoldásához tartson. Ezeket a definíciókat és tételeket például az [1] könyvben lehet megtalálni.

Definíció: Egy B mátrix spektrálsugara a B mátrix legnagyobb abszolút sajátértéke:

$$\rho(B) = \max |\lambda|,$$

ahol λ az B mátrix sajátértékeit jelöli.

Definíció: Tegyük fel hogy X teljes metrikus tér és $\phi : X \rightarrow X$ függvényre teljesül, hogy

$$d(\phi(x), \phi(y)) \leq q \cdot d(x, y), \quad \forall x, y \in X, \quad 0 < q < 1,$$

akkor ϕ kontrakció és q a kontrakciós konstans.

Tétel: (Banach-fixponttétel) Legyen X teljes metrikus tér és $\rho : X \rightarrow X$ kontrakció. Ekkor ρ -nak létezik egyetlen x^* -gal jelölt fixpontja és tetszőleges $x_0 \in X$ kezdőpont esetén az $x_n := \rho(x_{n-1})$ iteráció konvergens, és $\lim_{n \rightarrow \infty} x_n = x^*$. Sőt, érvényes a $d(x^*, x_m) \leq \frac{1}{1-q} \cdot d(x_1, x_0)q^m$ becslés, ahol q a kontrakciós konstans.

Tétel: Ha $B \in \mathbb{C}^{n \times n}$ és $\|\cdot\|$ egy indukált mátrixnorma, akkor igaz, hogy

$$\rho(B) \leq \|B\|.$$

Megfordítva: minden B mátrixhoz és minden $\epsilon > 0$ létezik olyan vektornorma, ami által indukált mátrixnorma esetén igaz, hogy

$$\|B\| \leq \rho(B) + \epsilon.$$

Tétel: Az iteráció tartani fog az egyenletrendszer megoldásához minden kiinduló vektor esetén akkor és csak akkor, ha az iterációs mátrix spektrálsugara kisebb, mint egy.

Bizonyítás

\Rightarrow

Az előző tétel szerint, ha $\rho(B) < 1$, akkor létezni fog olyan mátrixnorma, melyre ha ϵ -t elég kicsire választjuk, igaz lesz, hogy

$$\|B\| \leq \rho(B) + \epsilon < 1,$$

azaz

$$\|B\| < 1$$

Legyen

$$\varphi(x) = Bx + b.$$

Akkor felírhatjuk következőket, hogy

$$\|\varphi(x) - \varphi(y)\| = \|B(x - y)\| \leq \|B\| \cdot \|x - y\|.$$

De mivel $\|B\| < 1$, ezért φ kontrakció \mathbb{R}^n -ben. Azaz, a Banach fixponttétel szerint létezni fog egy fixpont:

$$\exists! \quad x^* \in \mathbb{R}^n : Bx^* + b = x^*.$$

Ezért

$$x^{(m+1)} = Bx^{(m)} + b = \varphi(x^{(m)})$$

esetén $x^{(m+1)}$ és $x^{(m)}$ tartani fog fixponthoz, azaz az egyenletrendszer megoldásához.

\Leftarrow Indirekt tegyük fel, hogy $\rho(B) > 1$ és ennek ellenére konvergálni fog az egyenletrendszer megoldásához. Akkor létezni fog B -nek egy olyan sajátértéke, ami legalább egy. Jelölje az ehhez a sajátértékhez tartozó sajátvektort s . Akkor az $Ax = f$ egyenletrendszerben úgy válasszuk meg f -et, hogy mikor átalakítjuk $x^{(m+1)} = Bx^{(m)} + b$ iterációvá, akkor $b = s$ jöjjön ki. Így:

$$x^{(0)} = s$$

$$x^{(1)} = Bx^{(0)} + b = Bs + s = \lambda s + s$$

$$x^{(2)} = Bx^{(1)} + b = B(\lambda s + s) + s = \lambda^2 s + \lambda s + s$$

\vdots

$$x^{(m+1)} = \left(\sum_{k=0}^{m+1} \lambda^k \right) s \rightarrow \infty,$$

ami ellentmondás.

□

3. fejezet

Alapvető iterációs módszerek: Jacobi-és Gauss-Seidel-iterációk

Bontsuk fel az A mátrixot egy alsó (U), egy felső (L) és egy diagonális (D) mátrix összegére. Szemléltetve:

$$U = \begin{pmatrix} 0 & a_{12} & \dots & a_{1n} \\ 0 & 0 & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & a_{nn} \end{pmatrix}, L = \begin{pmatrix} 0 & 0 & \dots & 0 \\ a_{21} & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ a_{31} & a_{32} & \dots & 0 \end{pmatrix}, D = \begin{pmatrix} a_{11} & 0 & \dots & 0 \\ 0 & a_{22} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & a_{nn} \end{pmatrix}$$

Feltesszük továbbá azt is, hogy a diagonális elemek az A mátrixban nem nullák. (Ha esetleg olyan A mátrix lenne, aminek a diagonálisában egy vagy több nulla áll, akkor két vagy több sor cseréjével ez kiküszöbölhető. Ennél fogva D inverze is létezik. Az alábbi két módszerről a [3],[1] és a [2] könyvekben lehet részletesebben olvasni.

3.1. Jacobi módszer

A fent leírt három mátrix segítségével az $Ax = f$ egyenletet a következő formába lehet írni:

$$x = -D^{-1}(L + U)x + D^{-1}f,$$

amiből az iteráció:

$$x^{(m+1)} = -D^{-1}(L + U)x^{(m)} + D^{-1}f,$$

így egy sor az iterációból:

$$x_j^{(m+1)} = \sum_{k=1, k \neq j}^n \frac{a_{jk}}{a_{jj}} x_k^{(m)} + \frac{f_j}{a_{jj}}, \quad j = 1, \dots, n$$

Tétel: Tegyük fel, hogy az $A = (a_{jk})$ mátrix kielégíti a következő feltételek valamelyikét:

$$q_\infty := \max_{j=1, \dots, n} \sum_{k=1, k \neq j}^n \left| \frac{a_{jk}}{a_{jj}} \right| < 1, \quad \text{vagy}$$

$$q_1 := \max_{k=1, \dots, n} \sum_{j=1, j \neq k}^n \left| \frac{a_{jk}}{a_{jj}} \right| < 1, \quad \text{vagy}$$

$$q_2 := \left(\sum_{j,k=1, j \neq k}^n \left| \frac{a_{jk}}{a_{jj}} \right|^2 \right)^{1/2} < 1.$$

Ekkor a Jacobi-módszer konvergál $\forall f \in \mathbb{R}^n$ jobboldal és $\forall x^{(0)} \in \mathbb{R}^n$ kezdővektor esetén az $Ax = f$ egyértelmű megoldásához.

3.2. Gauss-Seidel-iteráció

A Jacobi módszerrel ellentétben a következőképp is átalakíthatjuk az $Ax = f$ egyenletet:

$$(D + L)x = -Ux + f,$$

ami ekvivalens azzal, hogy

$$x = -(D + L)^{-1}Ux + (D + L)^{-1}f,$$

amiből az iteráció:

$$x^{(m+1)} = -(D + L)^{-1}Ux^{(m)} + (D + L)^{-1}f$$

tetszőleges $x^{(0)}$ kiinduló vektorral. Így egy sor az iterációból:

$$x_j^{(m+1)} = - \sum_{k=1}^{j-1} \frac{a_{jk}}{a_{jj}} x_k^{(m+1)} - \sum_{k=j+1}^n \frac{a_{jk}}{a_{jj}} x_k^{(m)} + \frac{f_j}{a_{jj}}, \quad j = 1, \dots, n.$$

A Gauss-Seidel iterációnak egyik jellemzője, hogy $x_j^{(m+1)}$ kiszámításakor már felhasználjuk $x_i^{(m)}$, $i < j$ kiszámolt új közelítéseket. Míg a Jacobi módszer végig az m -edik közelítéseket

használja az $x^{(m+1)}$ helyett. A Jacobi módszernél minden lépésben az összes régi m közelítő értéket tároljuk és csak aztán írjuk felül az új adatokkal, mikor azokat kiszámoltuk. Addig a Gauss-Seidel iterációban, amint kiszámoltuk az új közelítő értékét, már nincs szükségünk a régre, mert a következő közelítő érték számolásánál már az előző lépésben kiszámolt értéket használjuk fel. Ez számítógépek esetében helytakarékos megoldás. A következő tétel elnevezése Sassenfeld-kritérium, és a Gauss-Seidel iterációval kapcsolatban mondja ki a konvergenciához szükséges feltételeket (lásd [1]).

Tétel: *Tegyük fel hogy az B mátrix eleget tesz a következő feltételnek:*

$$p := \max_{j=1 \dots n} p_j < 1,$$

ahol a p_j számokat a:

$$p_1 = \sum_{k=2}^n \left| \frac{a_{1k}}{a_{jj}} \right|$$

$$p_j = \sum_{k=1}^{j-1} \left| \frac{a_{jk}}{a_{jj}} \right| p_k + \sum_{k=j+1}^n \left| \frac{a_{jk}}{a_{jj}} \right|, \quad j = 2 \dots n$$

rekurzióval határozzuk meg. Akkor a Gauss-Seidel iteráció minden lépésben konvergál a megoldáshoz.

4. fejezet

Relaxációs módszerek

A relaxációs módszerek ötlete arról szól, hogy egy paraméter bevezetésével igyekszünk gyorsítani az iterációs eljárásán. Ezt úgy oldjuk meg, hogy igyekszünk ezzel a paraméterrel a mátrix spektrálsugarát minimálisra csökkenteni. Ebben a fejezetben az előző két alapvető iterációs eljárás, a Jacobi és a Gauss-Seidel relaxációs paraméter bevezetésével vett javítását nézem meg. Nevezetesen a JOR és a SOR módszert és a SOR módszert tridiagonális mátrixokra.

4.1. JOR módszer

A Jacobi módszer általánosítása egy relaxációs módszerré az ω paraméter bevezetésével a következőképpen néz ki:

$$x^{(m+1)} = x^{(m)} + \omega D^{-1}(f - Ax^{(m)}),$$

amiből megkaphatjuk a módszer iterációs mátrixa a $B(\omega) := I - \omega D^{-1}A$ lesz. Az iteráció részletesen egy sorra kiírva:

$$x_j^{(m+1)} = x_j^{(m)} + \frac{\omega}{a_{jj}} \left[f_j - \sum_{k=1}^n a_{jk} x_k^{(m)} \right], \quad j = 1, \dots, n.$$

Vegyük észre, hogy a JOR módszer $\omega = 1$ esetén azonos lesz a Jacobi-módszerrel. A következő tétel az ω paraméter optimális megválasztásával kapcsolatban mondja ki az alábbiakat (lásd [1]).

Tétel: Tegyük fel, hogy az iterációs mátrixban a sajátértékek valósak és teljesül rá még a $\rho(B(\omega)) < 1$ feltétel is. Akkor az iterációs mátrix spektrálsugara következő ω megválasztásával minimális lesz:

$$\omega_{opt} = \frac{2}{2 - \lambda_{max} - \lambda_{min}},$$

amivel az iterációs mátrix spektrálsugara

$$\rho(I - \omega_{opt}D^{-1}A) = \frac{\lambda_{max} - \lambda_{min}}{2 - \lambda_{max} - \lambda_{min}}$$

lesz, ahol λ_{max} és λ_{min} az iterációs mátrix legnagyobb illetve a legkisebb sajátértéke. Amennyiben teljesül az a feltétel, hogy $-\lambda_{max}$ és λ_{min} nem egyenlőek, akkor a JOR módszer gyorsabb lesz, mint a Jacobi módszer.

Bizonyítás $\omega > 0$ esetén a $Ax = \lambda x$ egyenletet felírhatjuk a következő alakban is

$$((1 - \omega)I + \omega A)x = ((1 - \omega) + \omega\lambda)x.$$

Mivel A sajátértékei valósak, ezért $((1 - \omega)I + \omega A)$ sajátértékei is valósak lesznek. Amiből következik, hogy a legkisebb, illetve a legnagyobb sajátértéke $(1 - \omega) + \omega\lambda_{min}$ illetve $(1 - \omega) + \omega\lambda_{max}$ lesz. Nyilvánvaló, hogy a spektrálsugár akkor lesz a legkisebb, amikor a legnagyobb és a legkisebb sajátértékének ellentétes az előjele és az abszolút értékük megegyezik, vagyis:

$$(1 - \omega) + \omega\lambda_{min} = -(1 - \omega) - \omega\lambda_{max}.$$

Amiből ω -t kifejezve megkapjuk a tétel elején állított képletet az ω optimális értékére:

$$\omega_{opt} = \frac{2}{2 - \lambda_{max} - \lambda_{min}}.$$

□

4.2. SOR módszer

A Gauss-Seidel módszert egy ω paraméter bevezetésével gyorsítani lehet. Ezzel az iteráció következőképpen fog kinézni:

$$x^{(m+1)} = x^{(m)} + \omega D^{-1} [f - Lx^{(m+1)} - (D + U)x^{(m)}], \quad m = 1, 2, \dots$$

Így a SOR módszer iterációs mátrixa

$$B(\omega) := (D + \omega L)^{-1} [(1 - \omega)D - \omega U].$$

A következő tétel az ω paraméter megválasztásával kapcsolatban mond ki egy szükséges feltételt. (lásd [1])

Tétel: (Kahan) *Ha a SOR módszer konvergens, akkor az ω paraméter értéke a $(0, 2)$ intervallumban van.*

Bizonyítás Legyenek az iterációs mátrix sajátértékei: (η_1, \dots, η_n) . Mivel a karakterisztikus polinom gyökei, azaz teljesítik a következő feltételt:

$$\prod_{j=1}^n \eta_j = \det B(\omega).$$

Mivel a determináns multiplikatív és $(D + \omega L), (1 - \omega)D - \omega U$ tridiagonális mátrixok, ezért:

$$\prod_{j=1}^n \eta_j = \det(D + \omega L)^{-1} \det[(1 - \omega)D - \omega U] = (1 - \omega)^n.$$

Ebből következik, hogy biztosan igaz lesz, hogy

$$\rho(B(\omega)) \geq |1 - \omega|,$$

És mivel korábban láttuk, ha az $\rho(B(\omega)) < 1$, akkor az iteráció konvergens lesz, ezért ez csak akkor teljesül, hogy ha az ω paraméter a $(0, 2)$ intervallum eleme.

□

A következő tételt Ostrowski-tétel néven ismert, ami a bizonyítással együtt megtalálható az [1] könyvben. Ez a tétel már szükséges és elégséges feltételt ad a konvergenciára. Igaz, hogy a mátrixnak további feltételeknek kell eleget tennie.

Tétel: *Ha az A mátrix pozitív definit és önadjungált, akkor a SOR módszer minden kiinduló vektor és minden $0 < \omega < 2$ paraméter esetén az $Ax = f$ egyenletrendszer megoldásához fog konvergálni.*

Bizonyítás Legyen η az iterációs mátrix sajátértéke. Akkor a következő egyenlőséget írhatjuk fel:

$$2[(1 - \omega)D - \omega U]x = 2\eta(D + \omega L)x,$$

ahol x az η -hoz tartozó sajátvektor. A fenti egyenlet bal oldalát át tudjuk írni

$$(2 - \omega)D - \omega A - \omega(U - L)$$

alakra, míg a jobboldalra

$$(2 - \omega)D + \omega A - \omega(U - L).$$

Ezeket visszaírva a kezdő egyenletünkbe

$$[(2 - \omega)D - \omega A - \omega(U - L)]x = \eta[(2 - \omega)D + \omega A - \omega(U - L)]x,$$

melyben x -el skalárisan szorozva és η -t kifejezve kapjuk:

$$\frac{(2 - \omega)(Dx, x) - \omega(Ax, x) - i\omega i(Ux - Lx, x)}{(2 - \omega)(Dx, x) + \omega(Ax, x) - i\omega i(Ux - Lx, x)} = \eta.$$

Mivel A pozitív definit, ezért (Ax, x) és (Dx, x) nagyobb lesz, mint nulla. Az $U - L$ ferdén szimmetrikus is, ezért $i(Ux - Lx, x)$ ráadásul valós lesz. Ahhoz, hogy az iteráció konvergens legyen, $|\eta|$ -nak kisebbnek kell lennie, mint egy. Vagyis

$$|(2 - \omega)(Dx, x) - \omega(Ax, x)| > |(2 - \omega)(Dx, x) + \omega(Ax, x)|$$

Ez pedig az $0 < \omega < 2$ érték esetében lesz igaz. Amiből következik, hogy $|\eta| < 1$, amiből már az látszik, hogy $\max |\eta| < 1$. Ebből kifolyólag igaz lesz, hogy:

$$\rho(\eta) < 1.$$

Amivel bizonyítottuk a tételt. □

4.3. Relaxációs módszerek tridiagonális mátrixokra

Az optimális relaxációs paraméter tulajdonképpen nem mást, mint az olyan ω paraméter, melyre az iterációs mátrix spektrálsugara a legkisebb lesz. Aminek a számítása nehéz, egy-két kivételtől eltekintve. Mindenesetre érdemes az ω paraméter optimális értékét megkeresni, mivel nagy különbségek lehetnek két különböző paraméterrel végzett iteráció futási

ideje közt. Amikre később, pár mintafeladat képében példát is látunk. Itt speciálisan a tridiagonális mátrixokkal kapcsolatban fogom megnézni, hogy mi a SOR módszert optimális paramétere. Evvel kapcsolatban a Young tétel mondja ki az optimális paraméter értékét. De előtte egy definíciót és egy tételt mondok ki, ami az [1] könyv alapján van, és amik szükségesek a Young tétel megértéséhez és bizonyításához.

Definíció: Az A mátrixot, melynek diagonálisában nincsen nulla, konzisztensen rendezettnek hívunk, ha

$$C(\alpha) := -\alpha D^{-1}L - \frac{1}{\alpha} D^{-1}U, \quad (\alpha \in \mathbb{C})$$

sajátértékei nem függenek α -tól.

Tétel: Nem nulla diagonális elemekkel rendelkező tridiagonális mátrixok konzisztensen rendezettek.

Bizonyítás Legyen az

$$D(\alpha) = \begin{pmatrix} 1 & * & \dots & 0 \\ * & \alpha^2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \alpha^{n-1} \end{pmatrix}.$$

Akkor a következő egyenlőség igaz lesz, hogy

$$D(\alpha)C(1)D^{-1}(\alpha) = C(\alpha).$$

amiből következik, hogy minden $C(\alpha)$ hasonló $C(1)$ -el, ezért ugyanaz lesz a sajátértékük.

□

Tétel: (Young) Tegyük fel, hogy A konzisztensen rendezett és a Jacobi iterációs mátrixának sajátértékei valósak és spektrálsugara kisebb, mint egy. Akkor a SOR módszer minden $0 < \omega < 2$ érték esetén konvergens. Az ω optimális értéke

$$\omega_{opt} = \frac{2}{1 + \sqrt{1 - \Lambda^2}} \geq 1,$$

ahol Λ a Jacobi iterációs mátrix spektrálsugara. A SOR módszer iterációs mátrixának a spektrálsugara pedig:

$$\rho [B(\omega_{opt})] = \frac{1 - \sqrt{1 - \Lambda^2}}{1 + \sqrt{1 - \Lambda^2}}.$$

Bizonyítás Az

$$(I + \omega D^{-1}L) [\mu I - B(\omega)]$$

mátrixot felírhatjuk

$$\mu(I + \omega D^{-1}L) - D^{-1}[(1 - \omega)D - \omega U]$$

alakban is, amit át tudunk írni

$$(\mu + \omega - 1)I + \sqrt{\mu}\omega(\sqrt{\mu}D^{-1}L + \frac{1}{\sqrt{\mu}}D^{-1}U)$$

alakra. Tudjuk, hogy $I + \omega D^{-1}L$ determinánsa nem nulla, ezért láthatjuk, hogy $\mu \neq 0$ $B(\omega)$ sajátértéke, akkor és csak akkor, ha

$$\lambda := \frac{\mu + \omega - 1}{\sqrt{\mu}\omega}$$

sajátértéke a

$$-\sqrt{\mu}D^{-1}L - \frac{1}{\sqrt{\mu}}D^{-1}U$$

mátrixnak. Mivel A konzisztensen rendezett, ezért μ akkor és csak akkor sajátértéke a $B(\omega)$ mátrixnak, ha λ sajátértéke az $-D^{-1}(L + U)$ mátrixnak. Megoldva a

$$\lambda\sqrt{\mu}\omega = \mu + \omega - 1$$

másodfokú egyenletet, kapjuk, hogy

$$\mu = \left(\frac{\omega\lambda}{2} \pm \sqrt{\frac{\omega^2\lambda^2}{4} + 1 - \omega} \right)^2.$$

Ha $\alpha = -1$, akkor a

$$C(-1) := D^{-1}L + D^{-1}U$$

képletből nyilvánvaló, hogy ha λ sajátértéke a $-D^{-1}(L + U)$ mátrixnak, akkor $-\lambda$ is sajátértéke. Mivel mi csak $B(\omega)$ spektrálsugarára vagyunk kíváncsiak, ezért nézhetjük

$$\mu = \left(\frac{\omega|\lambda|}{2} + \sqrt{\frac{\omega^2\lambda^2}{4} + 1 - \omega} \right)^2$$

képletet. Mivel $|\alpha| < 1$ a feltétel szerint, ezért az

$$\omega^2\lambda^2 - 4\omega + 4 = 0$$

egyenletnek két valós megoldása lesz. Ennek két valós megoldásból csak az egyik fog a $(0, 2)$ intervallumba beleesni, nevezetesen:

$$\omega_0(\lambda) = \frac{2}{1 + \sqrt{1 - \lambda^2}} \geq 1.$$

Ami azt jelenti, hogy

$$\omega^2 \lambda^2 - 4\omega + 4 \geq 0 \quad \text{ha } 0 < \omega \leq \omega_0(\lambda).$$

Ebből

$$|\mu(\omega)| = \left(\frac{\omega|\lambda|}{2} + \sqrt{\frac{\omega^2 \lambda^2}{4} + 1 - \omega} \right)^2, \quad \text{ha } 0 < \omega \leq \omega_0(\lambda).$$

Ha $\omega_0(\lambda) < \omega < 2$ akkor komplex sajátértékek vannak és

$$|\mu(\omega)| = \omega - 1.$$

Előző kettőből látható, hogy $|\mu(\omega)|$ monoton nem csökken a $|\lambda|$ -ban. Ezért az iterációs mátrix spektrálsugara:

$$\left(\frac{\omega\Lambda}{2} + \sqrt{\frac{\omega^2 \Lambda^2}{4} + 1 - \omega} \right)^2,$$

ha $0 < \omega \leq \omega_0(\Lambda)$, és

$$\omega - 1,$$

ha $\omega_0(\Lambda) < \omega < 2$. Így az

$$f(\omega) := \frac{\omega\Lambda}{2} + \sqrt{\frac{\omega^2 \Lambda^2}{4} + 1 - \omega}$$

függvény nulla helyen felvett értéke egy, a deriváltja pedig:

$$f'(\omega) = \frac{\Lambda}{2} + \frac{\omega\Lambda^2 - 2}{2\sqrt{\omega^2 \Lambda^2 + 4 - 4\omega}} < 0.$$

Ez utóbbi a

$$\Lambda^2(4 - 4\omega + \omega^2 \Lambda^2) < (2 - \omega\Lambda^2)^2$$

egyenlőtlenségből következik. Ezért a spektrálsugár szigorúan monoton csökken a $0 < \omega \leq \omega_0$ esetén és szigorúan monoton nő az $\omega_0 < \omega < 2$ intervallumon. Mivel az iterációs mátrix spektrálsugara nullában és kettőben szintén egy, ezért ebből következik, hogy az iterációs mátrix spektrálsugara nem lehet nagyobb, mint egy a $(0, 2)$ intervallumon. A minimumát,

ami az ω paraméter optimális értéke, $\omega_0 = \omega_{opt}(\Lambda)$ helyen veszi fel $\omega_0(\Lambda) - 1$ értékkel, azaz

$$\rho[B(\omega_{opt})] = \frac{1 - \sqrt{1 - \Lambda^2}}{1 + \sqrt{1 - \Lambda^2}}$$

□

5. fejezet

Az eddigi módszerek kipróbálása különböző mátrixokon

Ebben a fejezetben, a Matlab program segítségével igyekszek különböző mátrixokon végrehajtani az előbbi fejezeteken vett numerikus módszereket és az eredményeket összehasonlítani. A programokhoz a [2] könyv algoritmusait vettem alapul. Mivel a módszerek idejét is mérni akarom, ezért fontos feljegyezni a gépem konfigurációját, mivel ez is nagyban befolyásolja egyes módszerek futási idejét:

Windows XP SP 3
Genuine Intel(R) CPU
T1600 @ 1.66Ghz
3Gb RAM

5.1. Tridiagonális mátrix

Itt az eddigi módszereket tesztelem a $\text{tridiag}(-1, 2, -1)$ mátrixra. Jelen esetben egy 10×10 -es mátrixot vettem. Előtte viszont megnéztem, hogy a Sassenfeld-kritériumot teljesíti-e

az alábbi mátrix:

$$\begin{bmatrix} 2 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 2 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 2 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 2 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 2 \end{bmatrix}.$$

Első lépésben megállapítottam, hogy

$$p_1 = \frac{1}{2},$$

majd kiszámoltam a p_j számokat, amikre a következő eredményt kaptam:

| | | | | |
|-----------------------|-------------------------|-------------------------|-------------------------|------------------------------|
| $p_1 = \frac{1}{2}$ | $p_2 = \frac{3}{4}$ | $p_3 = \frac{7}{8}$ | $p_4 = \frac{15}{16}$ | $p_5 = \frac{31}{32}$ |
| $p_6 = \frac{63}{64}$ | $p_7 = \frac{127}{128}$ | $p_8 = \frac{255}{256}$ | $p_9 = \frac{511}{512}$ | $p_{10} = \frac{1023}{1024}$ |

Fentiekből megállapítottam, hogy $\max p_j < 1$, azaz teljesíti a tridiag(-1,2,-1) mátrix a kritériumot és a Gauss-Seidel módszer konvergálni fog a megoldáshoz. A tridiag(-1,2,-1) mátrixok esetében igaz lesz, hogy a p_n értékét a

$$p_n = 1 - \frac{1}{2^n}$$

képlettel könnyedén ki lehet számolni. Azaz igaz lesz rájuk nézve a Sassenfeld-kritérium (lásd [1]).

Utána az iterációk előtt a Jacobi-iterációhoz vett kritériumot is ellenőriztem. Itt a q_∞ esetre néztem meg, és megállapítottam, hogy a tridiag(-1,2,-1) mátrix teljesíti a q_∞ -hez kapcsolódó feltételt, emiatt a Jacobi is konvergálni fog a megoldáshoz.

Fentiek után kiinduló vektornak a csupa nulla vektort, míg az f vektornak a csupa egyes vektort választottam. A lépésszámot m jelöli, míg a futási időt t . Elsőnek a Gauss-

eliminációt alkalmaztam, így az egyenletrendszer pontos megoldása:

| | |
|----------|---------|
| x_1 | 5.0000 |
| x_2 | 9.0000 |
| x_3 | 12.0000 |
| x_4 | 14.0000 |
| x_5 | 15.0000 |
| x_6 | 15.0000 |
| x_7 | 14.0000 |
| x_8 | 12.0000 |
| x_9 | 9.0000 |
| x_{10} | 5.0000 |

A pontos eredmények kiszámítása után elkezdtem az iterációs módszereket, elsőnek a Jacobi iterációval kezdtem, ami a következő eredményeket adta:

| x_j | $m = 10$ | $m = 20$ | $m = 30$ | $m = 40$ | $m = 50$ |
|----------|----------|----------|----------|----------|----------|
| x_1 | 2.3291 | 3.2331 | 3.8312 | 4.2270 | 4.4888 |
| x_2 | 3.8623 | 5.6070 | 6.7567 | 7.5165 | 8.0189 |
| x_3 | 4.8359 | 7.2598 | 8.8646 | 9.9264 | 10.6286 |
| x_4 | 5.3662 | 8.2923 | 10.2257 | 11.5041 | 12.3494 |
| x_5 | 5.6143 | 8.7905 | 10.8933 | 12.2841 | 13.2039 |
| x_6 | 5.6143 | 8.7905 | 10.8933 | 12.2841 | 13.2039 |
| x_7 | 5.3662 | 8.2923 | 10.2257 | 11.5041 | 12.3494 |
| x_8 | 4.8359 | 7.2598 | 8.8646 | 9.9264 | 10.6286 |
| x_9 | 3.8623 | 5.6070 | 6.7567 | 7.5165 | 8.0189 |
| x_{10} | 2.3291 | 3.2331 | 3.8312 | 4.2270 | 4.4888 |
| t | | | | | |
| * | 0.000298 | 0.000437 | 0.000461 | 0.000943 | 0.001001 |

Mivel túl sok mindent nem árul el a fenti táblázat, ezért úgy döntöttem, hogy a pontos

megoldás és a közelítés értékének a különbségét nézem.

| x_j | $m = 10$ | $m = 20$ | $m = 30$ | $m = 40$ | $m = 50$ |
|----------|----------|----------|----------|----------|----------|
| x_1 | 2.6709 | 1.7669 | 1.1688 | 0.7730 | 0.5112 |
| x_2 | 5.1377 | 3.3930 | 2.2433 | 1.4835 | 0.9811 |
| x_3 | 7.1641 | 4.7402 | 3.1354 | 2.0736 | 1.3714 |
| x_4 | 8.6338 | 5.7077 | 3.7743 | 2.4959 | 1.6506 |
| x_5 | 9.3857 | 6.2095 | 4.1067 | 2.7159 | 1.7961 |
| x_6 | 9.3857 | 6.2095 | 4.1067 | 2.7159 | 1.7961 |
| x_7 | 8.6338 | 5.7077 | 3.7743 | 2.4959 | 1.6506 |
| x_8 | 7.1641 | 4.7402 | 3.1354 | 2.0736 | 1.3714 |
| x_9 | 5.1377 | 3.3930 | 2.2433 | 1.4835 | 0.9811 |
| x_{10} | 2.6709 | 1.7669 | 1.1688 | 0.7730 | 0.5112 |
| t | | | | | |
| * | 0.000298 | 0.000437 | 0.000461 | 0.000943 | 0.001001 |

Ezen már látható, hogy a pontos megoldás és a közelítő megoldás közti különbség minden egyes iterációs lépéssel csökken, azaz az iteráció tart az egyenletrendszer megoldásához. A JOR módszer az $\omega = 1$ esetén a Jacobi módszerrel megegyezik, ezért elsőnek az ω paramétert 0,5-nek választottam, amire a következő eredményt kaptam:

| x_j | $m = 10$ | $m = 20$ | $m = 30$ | $m = 40$ | $m = 50$ |
|----------|----------|----------|----------|----------|----------|
| x_1 | 3.3136 | 2.6872 | 2.1879 | 1.7828 | 1.4528 |
| x_2 | 6.3468 | 5.1548 | 4.1983 | 3.4211 | 2.7880 |
| x_3 | 8.8498 | 7.2024 | 5.8682 | 4.7822 | 3.8972 |
| x_4 | 10.6266 | 8.6651 | 7.0625 | 5.7558 | 4.6908 |
| x_5 | 11.5467 | 9.4266 | 7.6847 | 6.2631 | 5.1043 |
| x_6 | 11.5467 | 9.4266 | 7.6847 | 6.2631 | 5.1043 |
| x_7 | 10.6266 | 8.6651 | 7.0625 | 5.7558 | 4.6908 |
| x_8 | 8.8498 | 7.2024 | 5.8682 | 4.7822 | 3.8972 |
| x_9 | 6.3468 | 5.1548 | 4.1983 | 3.4211 | 2.7880 |
| x_{10} | 3.3136 | 2.6872 | 2.1879 | 1.7828 | 1.4528 |
| t | | | | | |
| * | 0.000404 | 0.000395 | 0.000522 | 0.000556 | 0.000582 |

A JOR módszer $\omega = 0,5$ paraméterre gyengébben konvergál a pontos megoldáshoz, mint a Jacobi módszer. Most erre a mátrixra a Gauss-Seidel módszert néztem, amire a következő

eredményt kaptam a hibára:

| x_j | $m = 10$ | $m = 20$ | $m = 30$ | $m = 40$ | $m = 50$ |
|----------|----------|----------|----------|----------|----------|
| x_1 | 2.1246 | 0.9520 | 0.4171 | 0.1825 | 0.0798 |
| x_2 | 3.9353 | 1.7537 | 0.7680 | 0.3359 | 0.1469 |
| x_3 | 5.3118 | 2.3533 | 1.0302 | 0.4506 | 0.1971 |
| x_4 | 6.1721 | 2.7189 | 1.1898 | 0.5204 | 0.2276 |
| x_5 | 6.4790 | 2.8399 | 1.2422 | 0.5433 | 0.2376 |
| x_6 | 6.2437 | 2.7257 | 1.1919 | 0.5213 | 0.2280 |
| x_7 | 5.5230 | 2.4040 | 1.0510 | 0.4597 | 0.2010 |
| x_8 | 4.4116 | 1.9166 | 0.8379 | 0.3664 | 0.1603 |
| x_9 | 3.0308 | 1.3157 | 0.5751 | 0.2515 | 0.1100 |
| x_{10} | 1.5154 | 0.6578 | 0.2876 | 0.1258 | 0.0550 |
| t | | | | | |
| * | 0.000330 | 0.000500 | 0.000750 | 0.000834 | 0.000976 |

A Jacobi módszernél sokkal jobban konvergál a Gauss-Seidel. Most mivel a Gauss-Seidel megegyezik a SOR módszer $\omega = 1$ paraméterrel, ezért a első lépésben az $\omega = 0.5$ -nek választottam.

| x_j | $m = 10$ | $m = 20$ | $m = 30$ | $m = 40$ | $m = 50$ |
|----------|----------|----------|----------|----------|----------|
| x_1 | 3.2109 | 2.4741 | 1.9002 | 1.4545 | 1.1113 |
| x_2 | 6.0905 | 4.6923 | 3.6008 | 2.7548 | 2.1043 |
| x_3 | 8.4273 | 6.4875 | 4.9722 | 3.8013 | 2.9028 |
| x_4 | 10.0572 | 7.7275 | 5.9130 | 4.5170 | 3.4480 |
| x_5 | 10.8690 | 8.3242 | 6.3580 | 4.8526 | 3.7027 |
| x_6 | 10.8071 | 8.2409 | 6.2823 | 4.7908 | 3.6541 |
| x_7 | 9.8767 | 7.4954 | 5.7037 | 4.3462 | 3.3139 |
| x_8 | 8.1528 | 6.1600 | 4.6803 | 3.5642 | 2.7169 |
| x_9 | 5.7835 | 4.3555 | 3.3058 | 2.5163 | 1.9177 |
| x_{10} | 2.9812 | 2.2413 | 1.7002 | 1.2939 | 0.9860 |
| t | | | | | |
| * | 0.000393 | 0.000618 | 0.000783 | 0.001096 | 0.000968 |

A SOR módszernél is látszik, hogy 0.5-nek megválasztva az ω -t sokkal gyengébben konvergál a megoldáshoz, mint a SOR módszer egynek választott paraméterrel. Ezek után

megnéztem a fenti módszerek iterációs mátrixának a spektrálsugarát:

| | | | |
|------------|-------------------------------|------------|-------------------------------|
| <i>JOR</i> | <i>JOR</i> ($\omega = 0.5$) | <i>SOR</i> | <i>SOR</i> ($\omega = 0.5$) |
| 0.9595 | 0.9797 | 0.92906 | 0.9733 |

Látszik, hogy minél kisebb az iterációs mátrix spektrálsugara, annál jobban konvergál az egyenletrendszer megoldásához az iteráció. Ezért nagyon fontos lehet a relaxációs módszerek esetében az optimális paraméter megkeresése. Erre egy nagy mátrix fog nagyon jó rálátást mutatni, ezért az optimális paraméter számítását ott fogom elvégezni.

5.2. Iterációk közti különbségek

A fentiekből mintha az derülne ki, hogy van egy nagyon jó módszerünk, ami mellett a többi nem érdemes használni. Ezért ebben a részben igyekszem megmutatni, hogy léteznek olyan mátrixok, amikor más módszereket érdemes használni. Legyen a következő mátrix:

$$\begin{bmatrix} 3 & 0 & 4 \\ 7 & 4 & 2 \\ -1 & 1 & 2 \end{bmatrix}.$$

Ehhez az f értékei legyenek [7, 13, 2], amiből látszik, hogy így az egyenletrendszer megoldása a csupa egyesekből álló vektor lesz. Az eddig módszereket néztem meg erre a mátrixra:

| Gauss | Jacobi($m = 40$) | Gauss-Seidel($m = 40$) | JOR($\omega = 0,5$) | SOR($\omega = 0,5$) |
|---------------|--------------------|--------------------------|-----------------------|-----------------------|
| 1 | 71, 1312 | $-8096 \cdot 10^4$ | 0,9948 | 1 |
| 1 | -32, 3694 | $11132 \cdot 10^4$ | 1 | 1 |
| 1 | -15, 1771 | $-9614 \cdot 10^4$ | 0,9985 | 1 |
| t | | | | |
| 0.015378 | 0,000379 | 0,009978 | 0,000458 | 0,000399 |
| spektrálsugár | | | | |
| * | 1.1251 | 1.5833 | 0,8927 | 0.6400 |

Látható, hogy a Gauss-Seidel-és a Jacobi-iteráció, mivel a spektrálsugaruk nem kisebb egy-nél, nem fog pontos megoldáshoz konvergálni. Míg az ω -t jól megválasztva a spektrálsugár egynél kisebb lesz, ezért a JOR és a SOR módszer konvergálni fog az egyenletrendszer megoldásához.

A következő mátrix jó példa lesz arra nézve, hogy létezik olyan eset, amikor a JOR módszer gyorsabban konvergál a megoldáshoz, mint a SOR módszer:

$$\begin{bmatrix} 7 & 6 & 9 \\ 4 & 5 & -4 \\ -7 & -3 & 8 \end{bmatrix}$$

Ami mellett az $f = [22, 5, -2]$. Elvégezve rajtuk $\omega = 1$ -re az iterációkat, és a hibát kiírva kapjuk, hogy

| <i>JOR</i> | <i>SOR</i> |
|------------|------------|
| $m = 5$ | |
| 0,3327 | 0,4248 |
| 0,5518 | 0,808 |
| 0,0015 | 0,0687 |
| $m = 10$ | |
| 0,049 | 0,2253 |
| 0,0484 | 0,2451 |
| 0,0285 | 0,1052 |

Eredményből látszik, hogy a JOR gyorsabban konvergál az egyenletrendszer megoldásához, mint a SOR módszer. Ha megnézzük a spektrálsugarukat, akkor látszik, hogy a JOR módszer iterációs mátrixának spektrálsugara (0.6411) jóval kisebb, mint a SOR módszer iterációs mátrixának spektrálsugara (0.7746).

5.3. Iterációk összehasonlítása egy eddigieknél nagyobb méretű mátrix esetén

Nagy méretű mátrixnak azt a tridiagonális mátrixot választottam, amit legelsőnek használtam fel, csak most 100×100 -as méretben vettem, aminél már az ω optimális értékét is kiszámolom. Mivel ez már sokkal nagyobb méretű mátrix, mint az eddigiek, ezért itt a megoldási módszereket következőképp hasonlítom össze:

1. Gauss-eliminációval kiszámolom a pontos megoldást
2. A SOR és a JOR relaxációs iterációs módszerekkel $m = 10, 40, 100, 1000, 5000$ lépésre nézem a pontos megoldás, és a közelítő megoldás közti legnagyobb eltérést, hogyan változik.

Első lépésben a relaxációs paramétert $\omega = 1$ -nek választottam, amire a következő eredményt kaptam:

| $m = 10$ | | $m = 40$ | | $m = 100$ | | $m = 1000$ | | $m = 5000$ | |
|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|
| <i>JOR</i> | <i>SOR</i> | <i>JOR</i> | <i>SOR</i> | <i>JOR</i> | <i>SOR</i> | <i>JOR</i> | <i>SOR</i> | <i>JOR</i> | <i>SOR</i> |
| 1270 | 1265 | 1255 | 1235 | 1225 | 1175 | 810,47 | 500,22 | 117,1 | 10,4 |

Látszik, hogy a SOR módszer, $\omega = 1$ esetén sokkal gyorsabban tart az egyenletrendszer megoldásához. Ezek után kiszámoltam az optimális paraméterét a két iterációs megoldásnak. A SOR módszer esetében az optimális paraméter $\omega = 1,9397$. Így a SOR módszer eredménye:

| $m = 10$ | $m = 40$ | $m = 100$ | $m = 1000$ |
|----------|----------|-----------|------------|
| 1037 | 491.67 | 30.3 | 0 |

A SOR módszer optimális paramétert megválasztva sokkalta gyorsabban konvergál a megoldáshoz és már 1000 iterációs lépés után felveszi a pontos megoldás értékét. Ellentétben az $\omega = 1$ értékkel, ahol még 5000 lépés után is jelentős eltérések mutatkoznak a pontos megoldás és a közelítő megoldás között. Ennek magyarázata az iterációs mátrixok spektrálsugarának méretében található meg, ugyanis a JOR módszer spektrálsugara 0.9995 lett. A SOR ($\omega = 1$) esetében a spektrálsugár 0.9990, az optimális ω paraméterre nézve pedig 0.9397.

6. fejezet

Összegzés

Szakterdolgozatomban megvizsgáltam néhány alapvető módszert az egyenletrendszerek megoldására. A megemlített iterációk mellett még nagyon sokféle megoldó módszer létezik, amit használnak is a gyakorlatban. Ez annak a bizonyítéka, hogy tökéletes minden esetben alkalmazható iterációs módszer nem létezik. Azt is lehetett látni, hogy az iterációk gyorsaságát milyen mértékben befolyásolta az iterációs mátrix spektrálsugára. Hiába volt egy alatt ennek értéke, ha nagyon közel volt egyhez, viszonylag lassan konvergált a megoldáshoz. Mindenképp muszáj megjegyezni, hogy érdemes jól megválasztani az iterációt és a hozzá tartozó optimális paramétert. Nagy mátrix esetében lehetett látni, hogy jelentősen megtudja növelni az iteráció konvergenciájának a gyorsaságát, ha ezt az értéket kiszámoljuk, ez által csökkentve a iterációs mátrix spektrálsugarának a méretét.

7. fejezet

Köszönetnyilvánítás

Szeretném megköszönni témavezetőmnek, Kurics Tamásnak a szakdolgozatommal kapcsolatos tanácsait, fáradozásait. Mindezek mellett szeretném azt is megköszönni, hogy türelemmel viselte szakdolgozatom során elkövetett hibáimat és segített ezek kijavításában.

Nyilatkozat

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

Irodalomjegyzék

- [1] R. Kress: Numerical Analysis, Springer Verlag, 1988
- [2] R. Sacco, A. Quarteroni, F. Saleri: Numerical Mathematics, Springer, 2000
- [3] Stoyan Gisbert, Takó Galina: Numerikus módszerek I. Typotex, 2002