

EÖTVÖS LORÁND TUDOMÁNYEGYETEM
TERMÉSZETTUDOMÁNYI KAR

**AZ LR- ÉS QR-ALGORITMUS
MÁTRIX SAJÁTÉRTÉK-FELADATOK
MEGOLDÁSÁRA**

Szakdolgozat

Váradai Mónika

Matematika BSc,
Alkalmazott matematikus szakirány

Témavezető: Dr. Hegedűs Csaba, megbízott előadó

Numerikus Analízis Tanszék



Budapest, 2012

Tartalomjegyzék

1. Bevezetés	1
2. Fontos alapismeretek	3
2.1. Definíciók, egyszerűbb tételek	3
2.2. Hasonlósági transzformációk, Schur normálformák	5
2.3. Mátrixok függvényei, inverz iteráció	9
2.4. Felső Hessenberg-mátrixok	11
3. Az LR- és QR-algoritmus	13
3.1. Az LR -algoritmus	13
3.2. A QR -algoritmus	16
4. A QR-algoritmus változatai	19
4.1. Az explicit eltolásos QR -algoritmus	19
4.2. QR -algoritmus Hessenberg-mátrixokra	26
4.3. Az implicit eltolásos QR -algoritmus	27
4.4. Az implicit dupla eltolásos QR -algoritmus	30
4.5. Többszörös sajátértékek esete	35
Köszönetnyilvánítás	36
Irodalomjegyzék	37

1. fejezet

Bevezetés

A sajátértékek ismerete nem csak a matematikában, hanem a valós életben is fontos, ugyanis sok gyakorlati probléma vezet sajátérték-feladatra: $Ax = \lambda x$, amikor is egy mátrix sajátértékeit és sajátvektorait kell meghatározni. Ilyenek például a differenciálegyenletek, a szilárdtest fizika, a digitális jel- és képfeldolgozás, a kvantummechanika és a gráfelmélet tudományterületei, a stabilitási, rezgési és rezonancia problémák, valamint pl. a Google is ilyen sajátérték feladatot old meg az internetes oldalak rangsorolásakor.

A sajátértékeket és sajátvektorokat egyrészt meg lehet határozni a jól ismert $\det(A - \lambda I) = 0$ karakterisztikus egyenlettel, ugyanis a gyökök pontosan A sajátértékei. Ez azonban a gyakorlatban nem használható, - mert numerikus szempontból így nagyon előnytelen a sajátértékek és sajátvektorok meghatározása, - ezért ehelyett iterációs módszereket kell alkalmaznunk. Ezeket a sajátérték-meghatározási módszereket két nagy csoportba lehet osztani: a sajátértékeket egyenként, illetve a sajátértékeket egyszerre közelítő eljárásokra.

- A sajátértékeket egyenként közelítő eljárásokra példa a Rayleigh-hányados, vagy a hatványmódszer (utóbbi az abszolút értékben legnagyobb sajátérték és a hozzá tartozó sajátvektor meghatározására alkalmas).
- A sajátértékeket egyszerre közelítő eljárások közül a ma már klasszikusnak számító Jacobi-módszer a legegyszerűbb, de csak abban az esetben, ha nem túl nagy a mátrix (a módszer célja, hogy eljussunk a $\text{diag}(\lambda_i(A))$ mátrixhoz, azaz a főátlón kívüli elemeket kinullázzuk). Ebbe a csoportba sorolhatók a rangszámcsökkentő eljárások is.

A sajátértékek komplex síkon való elhelyezkedését is ismerjük, ugyanis tudjuk, hogy abszolút értékben a mátrix egyik sajátértéke sem nagyobb, mint a mátrix tetszőleges normája. Ezzel máris kapunk egy kört, ami tartalmazza a mátrix összes sajátértékét. A

Gersgorin-körök azonban még ennél is pontosabb tartományt tudnak kijelölni a sajátértékek elhelyezkedésére.

A szakdolgozatomban mátrixok sajátértékeinek meghatározásával kapcsolatos problémák elméleti hátterét vizsgálom. Fő témám az LR -, valamint a mintájára készült QR -algoritmus és konvergenciájuk. Így a fenti, példaként felhozott eljárásokra nem térek ki, csak azért említettem őket, hogy lássuk a sajátértékeket meghatározó eljárások sokféleségét. A dolgozatom első részében a későbbiekben szükséges alapokat ismertetem. Itt a már tanultakat röviden összefoglalom és kiegészítem kevésbé ismert jelölésekkel, illetve a továbbiakban nélkülözhetetlen állításokkal. A következő részben az LR - és QR -algoritmust mutatom be, végül a QR -algoritmus változatait vizsgálom. A konvergenciát csak két esetben bizonyítom - ha a sajátértékek abszolút értékben különbözőek, illetve, ha komplex konjugált gyökpárok vannak, - ugyanis a konvergencia vizsgálata a többi esetben meglehetősen körülményes és bonyolult.

Dolgozatom alapja a Stewart könyv [1] Eigensystems c. fejezete, azon belül is különösen hangsúlyosak a 15-17. leckék, így az ott szereplő eredményekre minden további jelzés nélkül hivatkozok.

2. fejezet

Fontos alapismeretek

2.1. Definíciók, egyszerűbb tételek

Ebben a részben olyan, már tanult definíciókat és tételeket mondok ki, melyek ismerete nélkülözhetetlen a továbbiakban. Ismétlésképpen néhány állítást is kimondok, amikre a későbbiekben szükségünk lesz.

1. Definíció. Egy A mátrix konjugált transzponáltja az $A^H = \bar{A}^T$ mátrix (sok helyen $*$ -gal is jelölik).

2. Definíció. Az U négyzetes komplex mátrix unitér mátrix, ha $U^H U = I$, azaz konjugált transzponáltja egyben inverze is. Valós esetben: U ortogonális, ha $U^{-1} = U^T$.

2.1. Állítás. Unitér mátrixok szorzata is unitér.

Bizonyítás: Tegyük fel, hogy A és B unitér mátrixok. Ekkor $(AB)^{-1} = B^{-1}A^{-1} = B^H A^H = (AB)^H$, azaz AB unitér mátrix. \blacklozenge

2.2. Állítás.

- Unitér mátrixok sajátértékei 1 abszolút értékűek.
- Minden unitér mátrix minden sorának és minden oszlopának $\|\cdot\|_2$ -es normája 1.
- Unitér mátrixszal szorozva egy vektor vagy mátrix $\|\cdot\|_2$ -es normája nem változik:

$$\|Ux\|_2 = \|x\|_2.$$

- Speciálisan: ortogonális mátrixok $\|\cdot\|_2$ -es normája 1.

3. Definíció. Az A mátrixot involutórius mátrixnak nevezzük, ha teljesül az $A^2 = I$ ($A^{-1} = A$) összefüggés.

4. Definíció. A P mátrixot projektornak vagy vetítő-mátrixnak (néhol: idempotens mátrixnak) nevezzük, ha $P^2 = P$, vagyis a mátrix minden hatványa önmaga.

2.3. Állítás. Minden projektor $A = I - 2P$ alakban involutórius mátrixot határoz meg.

Bizonyítás: $(I - 2P)(I - 2P) = I - 2P - 2P + 4P^2 = I - 4P + 4P = I$. ♦

2.4. Állítás. Minden involutórius mátrix $(I - A)/2$ alakban meghatároz egy projektort.

Bizonyítás: $(I - A)(I - A)/4 = (I - A - A + A^2)/4 = (I - 2A + I)/4 = (I - A)/2$. ♦

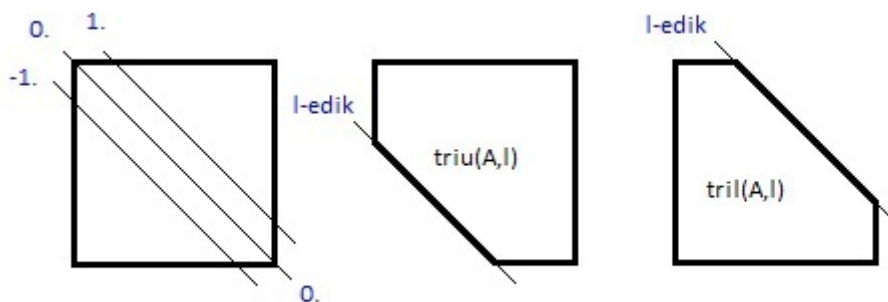
5. Definíció. Egy mátrix átlóit a következőképpen számozzuk: legyen i a sorindex, j pedig az oszlopindex. Ekkor az átlószám egyenlő $j - i$ -vel, tehát az oszlopindexből kivonjuk a sorindexet. Így a főátló száma 0, a közvetlenül alatta lévő átlóé -1, a közvetlenül fölötte lévőé pedig 1.

A következő jelölések a MATLAB programozási nyelv parancsai alapján kerültek bevezetésre a numerikus analízisben. Ezek olyan függvények, melyek mátrixhoz mátrixot rendelnek:

6. Definíció.

- A $\text{triu}(A, l)$ az A mátrixnak azon részét tartja meg, amely az l -edik átlótól fölfelé helyezkedik el (jobb felső sarok), a többi helyen zérus van.

- A $\text{tril}(A, l)$ a mátrix l -edik átlója alatti részt jelöli, beleértve az l -edik átlót is. A többi átlóban zérusok vannak.



7. Definíció. Az $u \in \mathbb{R}^n$ adott vektor esetén az

$$R(u) = I - \frac{2uu^T}{u^T u}$$

mátrixot az u vektorhoz tartozó Householder-mátrixnak, a vele való szorzást pedig Householder-transzformációnak vagy Householder-tükrözésnek nevezzük.

2.5. Állítás (Tulajdonságai):

- $R(u)$ szimmetrikus.
- $R(u)$ ortogonális (a szimmetria miatt ortogonalitás = involutórius, azaz $R(u)^2 = I$).
- $R(u)u = -u$.
- $R(u)x = x$ minden u -ra merőleges x esetén.

2.6. Tétel. Legyenek x és y $\|\cdot\|_2$ -ban azonos hosszúságú, de különböző vektorok. Ekkor az

$$u = \frac{x - y}{\|x - y\|_2}$$

vektorhoz tartozó Householder-transzformáció az x és y vektorokat egymásba tükrözi. A későbbiekben ezt így fogjuk jelölni: $R(x - y)x = y$.

8. Definíció. Az $\frac{x^H Ax}{x^H x}$ mennyiséget Rayleigh-hányadosnak nevezzük. Fontos szerepet játszik a sajátértékek és sajátvektorok elméletében és számítástechnikai gyakorlatában.

2.2. Hasonlósági transzformációk, Schur normálformák

Mátrixokkal való számítások legalapvetőbb technikája az, hogy a mátrixokat egyszerűbb alakra transzformáljuk, és ezzel leegyszerűsítjük a probléma megoldását. Ennek során fontos, hogy olyan transzformációkat hajtsunk végre, amelyek változatlanul hagyják azokat a dolgokat, amiket ki szeretnénk számolni. Sajátérték-feladatoknál az erre megfelelő transzformációkat hasonlósági transzformációknak nevezzük.

9. Definíció. Legyen S nonsinguláris mátrix. Ekkor az $S^{-1}AS$ mátrixra azt mondjuk, hogy hasonló az A mátrixhoz, és az $A \rightarrow S^{-1}AS$ hozzárendelést hasonlósági transzformációnak nevezzük.

Egy tetszőleges A mátrix hasonlósági transzformáltjának karakterisztikus polinomja megegyezik az eredeti mátrix karakterisztikus polinomjával:

$$\det(\lambda I - S^{-1}AS) \stackrel{S^{-1}S=I}{=} \det[S^{-1}(\lambda I - A)S] =$$

$$\det(S^{-1}) \det(\lambda I - A) \det(S) \stackrel{\det S^{-1} \det S = 1}{=} \det(\lambda I - A).$$

Ebből következik, hogy a hasonlósági transzformációk megőrzik a sajátértékeket azok multiplicitásával együtt (mert a sajátértékek a karakterisztikus polinom gyökei), a sajátvektorokat pedig kiszámítható módon transzformálják. Legyen $Ax = \lambda x$ és $u = S^{-1}x$. Így $(S^{-1}AS)u = \lambda u$. Ezért az $S^{-1}AS$ hasonlósági transzformáció az x jobb oldali sajátvektort $S^{-1}x$ -re alakítja át. Hasonlóan belátható, hogy az y^T bal oldali sajátvektor pedig $y^T S$ alakra transzformálódik.

10. Definíció. Legyen U unitér mátrix. Ekkor az $U^H A U$ transzformáció unitér hasonlósági transzformáció.

2.7. Tétel (Schur). Minden négyzetes (komplex) mátrix unitér hasonlósági transzformációval felső háromszög-alakra hozható, azaz $\forall n \times n$ -es A mátrixhoz $\exists n \times n$ -es U unitér és R felső háromszög mátrix, hogy $U^H A U = R$. Ezt Schur-normálformának nevezzük.

Bizonyítás: eredeti, [6] alapján.

Az A mátrix rendje szerinti teljes indukcióval bizonyítunk. Ez triviális $n = 1$ -re. Tegyük fel, hogy $n - 1$ -re már beláttuk, most szeretnénk n -re is bizonyítani. Legyen λ az $n \times n$ -es A mátrix egy sajátértéke, és legyen v a hozzá tartozó sajátvektor. Válasszuk v -t úgy, hogy $\|v\|_2 = 1$. Legyen U_1 egy olyan unitér mátrix, melynek első oszlopa v . Sok ilyen mátrix létezik, ugyanis vegyük \mathbb{C}^n bármelyik olyan ortonormált bázisát, melynek első tagja v , és U_1 olyan mátrix legyen, melynek oszlopai a bázis elemei. Ekkor U_1 felírható a következő alakban:

$$\mathbf{U}_1 = \begin{bmatrix} v & W \end{bmatrix},$$

ahol W az U_1 almátrixa. Mivel W oszlopai ortogonálisak v -re, $W^H v = 0$. Legyen $A_1 = U_1^H A U_1$. Ekkor

$$\mathbf{A}_1 = \begin{bmatrix} v^H \\ W^H \end{bmatrix} A \begin{bmatrix} v & W \end{bmatrix} = \begin{bmatrix} v^H A v & v^H A W \\ W^H A v & W^H A W \end{bmatrix}.$$

Mivel $A v = \lambda v$, ezért $v^H A v = \lambda$ és $W^H A v = \lambda W^H v = 0$. Legyen $\mathcal{A} = W^H A W$. Ekkor A_1 a következő alakú:

$$\mathbf{A}_1 = \left[\begin{array}{c|ccc} \lambda & * & \dots & * \\ \hline \underline{0} & & & \mathcal{A} \end{array} \right].$$

Mivel \mathcal{A} $(n - 1) \times (n - 1)$ -es mátrix, teljesül rá az indukciós feltétel, miszerint létezik egy olyan \mathcal{U}_2 unitér mátrix, és egy \mathcal{R} felső háromszög mátrix, hogy $\mathcal{R} = \mathcal{U}_2^H \mathcal{A} \mathcal{U}_2$. Definiáljuk U_2 $n \times n$ -es mátrixot a következőképpen:

$$\mathbf{U}_2 = \left[\begin{array}{c|ccc} 1 & 0 & \dots & 0 \\ \hline \underline{0} & & & \mathcal{U}_2 \end{array} \right].$$

Ekkor U_2 unitér, és

$$\mathbf{U}_2^H \mathbf{A}_1 \mathbf{U}_2 = \left[\begin{array}{c|ccc} \lambda & * & \dots & * \\ \hline \underline{0} & \mathcal{U}_2^H \mathcal{A} \mathcal{U}_2 & & \end{array} \right] = \left[\begin{array}{c|ccc} \lambda & * & \dots & * \\ \hline \underline{0} & & & \mathcal{R} \end{array} \right],$$

ami egy felső háromszög mátrix. Legyen ez a mátrix az R , és $U =: U_1 U_2$ (a 2.1. állítás alapján unitér). Így $T = U_2^H A_1 U_2 = U_2^H U_1^H A U_1 U_2 = U^H A U$. ♦

R átlójában az A mátrix sajátértékei vannak. Így, ha megtaláljuk azt az unitér hasonlósági transzformációt, ami az A -t felső háromszög-alakra hozza, akkor már ismerjük az A mátrix sajátértékeit is. Azonban ez nem konstruktív bizonyítás, mivel abból indulunk ki, hogy ismerünk egy sajátértéket. Mindemellett ez reményt kelt bennünk, hogy majd tudunk olyan algoritmust készíteni, ami unitér hasonló mátrixok sorozatával egy felső háromszög alakhoz konvergál (ez lesz a QR -algoritmus).

Bizonyítás: Householder-tükrözéssel, [2] alapján.

Jelölje $R(u)$ a Householder tükröző mátrixot. Legyen $x \neq e_1$ A egy normált sajátvektora, $\|x\|_2 = 1$ ($= \|e_1\|_2$). Ekkor $R(x - e_1) [x \ e_1] = [e_1 \ x]$ (2.6. alapján), azaz az x vektort e_1 -be viszi, e_1 -et pedig x -be. Feltéve, hogy $Ax = \lambda x$ teljesül,

$$R(x - e_1) A \underbrace{R(x - e_1) e_1}_{= x} = R(x - e_1) Ax = R(x - e_1) \lambda x = \lambda R(x - e_1) x = \lambda e_1.$$

Mivel a Householder-tükrözés involutórius (lásd 2.5.), azaz megegyezik inverzével, ezért az előbb hasonlósági transzformációt végeztünk, ahol az első oszlop a λ -szorosába ment át. Ezzel a felső háromszög alak első oszlopa, s így első sora is előállt. Ha az eljárást folytatjuk az eggyel kisebb méretű jobb alsó blokkra, végül eljutunk a felső háromszög alakra. ♦

Valós Schur forma

Függetlenül attól, hogy egy mátrix valós, a Schur formája lehet komplex is. Azonban a számítógépek megjelenésének kezdetén úgy gondolták, hogy amikor csak lehet, jobb kerülni a komplex mátrixokat (természetesen ma is dönthetünk úgy, hogy a valós aritmetika kényelmesebb, mint a komplex). Szerencsére a Schur formának létezik valós változata is.

2.8. Tétel. *Legyen $A \in \mathbb{R}^{n \times n}$. Ekkor létezik egy olyan U ortogonális mátrix, hogy az $U^T A U$ mátrix felső blokk-háromszög alakban jelenik meg:*

$$T = U^T A U = \begin{pmatrix} T_{11} & T_{12} & T_{13} & \dots & T_{1k} \\ 0 & T_{22} & T_{23} & \dots & T_{2k} \\ 0 & 0 & T_{33} & \dots & T_{3k} \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \dots & T_{kk} \end{pmatrix}$$

A diagonális blokkok vagy 1×1 -es méretűek - ekkor ezek éppen T valós sajátértékei; vagy 2×2 -esek - ebben az esetben T komplex konjugált sajátérték párait tartalmazzák. Ezt a T mátrixot nevezzük valós Schur formának.

1. Megjegyzés. *A 2×2 -es blokkok a következőképpen néznek ki:*

$$\begin{pmatrix} \alpha & \beta \\ -\beta & \alpha \end{pmatrix} \quad \text{vagy} \quad \begin{pmatrix} \alpha & -\beta \\ \beta & \alpha \end{pmatrix}$$

Bizonyítás: [1] alapján. Ezt az alakot a Schur normálformához hasonlóan bizonyíthatjuk, azzal a különbséggel, hogy a sajátértékek konjugált párait együttesen távolítjuk el. Vázlatosan, legyen $(\mu + i\nu, x + iy)$ egy komplex saját pár, így

$$A(x + iy) = (\mu + i\nu)(x + iy).$$

Könnyen igazolható, hogy x és y lineárisan függetlenek és

$$A \begin{pmatrix} x & y \end{pmatrix} = \begin{pmatrix} x & y \end{pmatrix} \begin{pmatrix} \mu & \nu \\ -\nu & \mu \end{pmatrix} \equiv \begin{pmatrix} x & y \end{pmatrix} M.$$

Figyeljünk, hogy M sajátértékeire teljesüljön $\mu \pm i\nu$. Legyen

$$\begin{pmatrix} V_1^T \\ V_2^T \end{pmatrix} \begin{pmatrix} x & y \end{pmatrix} = \begin{pmatrix} R \\ 0 \end{pmatrix}$$

az $\begin{pmatrix} x & y \end{pmatrix}$ QR -felbontása, tehát $V_1 = \begin{pmatrix} x & y \end{pmatrix} R^{-1}$. Így

$$\begin{aligned} \begin{pmatrix} V_1^T \\ V_2^T \end{pmatrix} AV_1 &= \begin{pmatrix} V_1^T \\ V_2^T \end{pmatrix} A \begin{pmatrix} x & y \end{pmatrix} R^{-1} = \\ &= \begin{pmatrix} V_1^T \\ V_2^T \end{pmatrix} \begin{pmatrix} x & y \end{pmatrix} MR^{-1} = \begin{pmatrix} RMR^{-1} \\ 0 \end{pmatrix}. \end{aligned}$$

Ebből következik, hogy

$$\begin{pmatrix} V_1^T \\ V_2^T \end{pmatrix} A \begin{pmatrix} V_1 & V_2 \end{pmatrix} = \begin{pmatrix} RMR^{-1} & V_1^T AV_2 \\ 0 & V_2^T AV_2 \end{pmatrix}.$$

Így A -t átalakítottuk egy valós blokk-háromszög alakra, amelyben a fő 2×2 -es blokk a sajátértékek egy komplex konjugált párját tartalmazza. Az eljárást folytatva a $V_2^T AV_2$ mátrixra, a kívánt alakra jutunk. ♦

2.3. Mátrixok függvényei, inverz iteráció

Mátrixok függvényei:

Ennek a résznek az ismertetését Stewart [1] nyomán tesszük.

Sok sajátértékekre kidolgozott algoritmus úgy alakítja át a mátrixot, hogy annak sajátértékei stratégiaileg jobban helyezkedjenek el.

- i. Ezek közül a legegyszerűbb transzformáció a koordináta-rendszer kezdőpontjának eltolása. Tegyük fel, hogy $Ax = \lambda x$. Ekkor $(A - \mu I)x = (\lambda - \mu)x$. Így tehát egy mátrix átlójából levont μ szám a sajátértékeket μ -vel fogja eltolni, azaz $\lambda_i - \mu$ értékűek lesznek, míg az eltolás a sajátvektorokat helyben hagyja.
- ii. Az $Ax = \lambda x$ egyenletből következik, hogy

$$A^2 x = A(Ax) = \lambda Ax = \lambda^2 x.$$

Így egy mátrix négyzetre emelése a sajátértékeket is négyzetre emeli. Általánoságban véve A^k sajátértékei A sajátértékeinek k -edik hatványai.

- iii. Ha $f(t) = \gamma_0 + \gamma_1 t + \dots + \gamma_n t^n$ egy polinom, akkor definiálhatjuk a következőt:

$$f(A) = \gamma_0 I + \gamma_1 A + \dots + \gamma_n A^n.$$

$f(A)$ és x összeszorzása, majd egyszerűsítés után kapjuk:

$$Ax = \lambda x \implies f(A)x = f(\lambda)x.$$

Így, ha A -nak egy polinomját képezzük, akkor a sajátértékek is eszerint a polinom szerint változnak.

- iv. Most tegyük fel, hogy A nonszinguláris és teljesül az $Ax = \lambda x$. Ekkor λ nem lehet nulla, mivel ellenkező esetben x nullvektor lenne (de egy sajátvektor nem lehet nullvektor), illetve azért, mert a sajátértékek szorzata egyenlő a determinánssal ($\det A = \prod_i \lambda_i$), ami most a feltevés miatt nem lehet zérus. Ebből következik, hogy

$$A^{-1} x = \lambda^{-1} x.$$

Így egy mátrix invertálásakor a mátrix sajátértékeinek reciprokait kapjuk.

- v. Legyen $h(t) = f(t)/g(t)$ egy racionális függvény és tegyük fel, hogy $g(A)$ nonszinguláris. Ekkor $h(A)$ -t így definiálhatjuk:

$$h(A) = g(A)^{-1} f(A).$$

A kapott eredményekből következik, hogy ha $Ax = \lambda x$, akkor

$$h(A)x = h(\lambda)x.$$

Így A sajátértékeit a h polinommal transzformáltuk.

- vi. Hatványsorral definiált függvényekre is igaz, hogy a sajátértékeket - amennyiben a konvergencia körön belülre esnek - az adott hatványsorral transzformálják. Ha például λ az A mátrix egy sajátértéke, akkor a következő hatványsor egyik sajátértéke e^λ lesz:

$$e^A \stackrel{\text{def}}{=} \sum_{k=0}^{\infty} \frac{A^k}{k!}.$$

Összefoglalva: egy mátrix tetszőleges függvénye a sajátértékeket a függvény által transzformálja, a sajátvektorokat pedig változatlanul hagyja.

Inverz iteráció

Hatványmódszerrel csak a spektrum (a mátrix sajátértékeinek összessége) szélein lévő egyszeres sajátértékeket kereshetjük eredményesen. Kellene tehát egy olyan módszer, amivel a spektrum belsejében található sajátértékeket is megkereshetjük.

Az előzőek alapján, hogy a mátrix sajátértékei jobb helyzetbe kerüljenek, transzformáljuk a mátrixot eltolással: $A - \lambda I$. A legjobb azonban az lesz, ha az eltolást inverzióval ötvözzük, azaz az $(A - \lambda I)^{-1}$ -re alkalmazzuk a hatványmódszert.

Legyen λ az A mátrix egy egyszeres λ_1 sajátértékének közelítőértéke. Az $(A - \lambda I)^{-1}$ mátrix sajátértékei

$$\frac{1}{\lambda_i - \lambda} \quad i = 1, 2, \dots, n.$$

Ahogy λ közelít λ_1 -hez, az $\frac{1}{\lambda_1 - \lambda}$ szám végtelenbe tart, míg az $\frac{1}{\lambda_i - \lambda}$ ($i \neq 1$) számok fix határértékekhez közelítenek. Ebből következik, hogy ahogy λ közelít λ_1 -hez, a hatványmódszer úgy fog egyre gyorsabban közelíteni a megfelelő sajátvektorhoz. A módszer hatásos még abban az esetben is, ha λ nem túl jó közelítése λ_1 -nek.

Ezt a módszert inverz hatványiterációnak nevezzük. Ekkor az iteráció egy lépésében az

$$x_{m+1} = (A - \lambda I)^{-1}x_m$$

vektort számítjuk ki. Az iteráció előnye a gyors konvergencia.

Összefoglalva: az inverz iteráció a λ -hoz legközelebbi sajátérték és a hozzá tartozó sajátvektor meghatározására alkalmas módszer, amennyiben nincs több ilyen távolságra lévő sajátérték.

2.4. Felső Hessenberg-mátrixok

11. Definíció (Hessenberg-mátrixok). Egy $n \times n$ -es mátrixot felső Hessenberg-mátrixnak nevezünk, ha $i > j + 1$ esetén $h_{ij} = 0$.

Egy 5×5 -ös felső Hessenberg-mátrix a következő alakú, ahol *-gal jelöljük a tetszőleges elemeket:

$$\mathbf{H} = \begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ 0 & * & * & * & * \\ 0 & 0 & * & * & * \\ 0 & 0 & 0 & * & * \end{bmatrix}$$

Röviden: H felső Hessenberg-mátrix, ha $\text{triu}(H, -1) = H$.

2.9. Tétel. Minden mátrix unitér hasonlósági transzformációval felső Hessenberg-alakra hozható.

Bizonyítás: [2] alapján.

Az első lépésben legyen $u_1 = \text{tril}(A, -1)e_1$, $\|u_1\|_2 = |\sigma_1|$. A hasonlósági transzformációt az $R(\text{tril}(A, -1)e_1 - \sigma_1 e_2) = R(u_1 - \sigma_1 e_2)$ Householder-mátrixszal végezzük, ahol u_1 előjelét úgy választjuk meg, hogy $\text{sign}(\sigma_1) := -e_2^T u_1$ teljesüljön a numerikusan stabil előjelválasztás miatt. Ekkor $R(u_1 - \sigma_1 e_2)A$ az első oszlopot $a_{11}e_1 + \sigma_1 e_2$ -be viszi.

Ha azonban $a_{21} = \dots = a_{n1} = 0$, akkor rögtön tovább is mehetünk a következő lépésre, vagy mondhatjuk azt, hogy ebben az esetben az egységmátrix legyen a tükröző mátrix.

A számunkra most érdektelen elemeket *-gal jelölve az előbbi tükrözés így néz ki:

$$\begin{pmatrix} a_{11} & * & * & * \\ a_{21} & * & * & * \\ \vdots & & & \end{pmatrix} \rightsquigarrow \begin{pmatrix} a_{11} & * & * & * \\ \sigma_1 & * & * & * \\ 0 & & & \\ \vdots & & & \\ 0 & & & \end{pmatrix}$$

Ugyanezzel a tükröző mátrixszal jobbról szorozva az első oszlop már nem fog változni: $R(u_1 - \sigma_1 e_2)e_1 = e_1$ és $e_1^T R(u_1 - \sigma_1 e_2) = e_1^T$, mert $R(u_1 - \sigma_1 e_2)$ első sora e_1^T és első oszlopa e_1 . Ezzel $A_2 = R(u_1 - \sigma_1 e_2)AR(u_1 - \sigma_1 e_2)$ első oszlopa már mutatja a Hessenberg-alakot.

A következő lépésben ez előbbieket alkalmazzuk A_2 eggyel kisebb méretű jobb alsó blokkjára, tehát $u_2 = \text{tril}(A_2, -1)e_2$, $\|u_2\|_2 = |\sigma_2|$ és a tükröző mátrix $R(u_2 - \sigma_2 e_3)$. Az eljárást tovább folytatva végül a kívánt felső Hessenberg-alakra jutunk. \blacklozenge

2. Megjegyzés. Az összes tükröző mátrix összeszorzása után a teljes unitér transzformáció mátrixa olyan, hogy az első sora és oszlopa e_1^T és e_1 . Ennek a ténynek külön jelentősége van az implicit eltolásos QR-módszernél.

2.10. Állítás. Ha A $n \times n$ -es felső Hessenberg-mátrix, B $n \times n$ -es felső háromszög mátrix, akkor AB és BA is felső Hessenberg-mátrix.

Bizonyítás: Legyen $B = [b_{ij}]$, $b_{ij} = 0$, ha $i > j$. Legyen $AB = [c_{ij}]$, ahol $c_{ij} = \sum_{k=1}^n a_{ik}b_{kj}$. Ha $i > k + 1$, akkor $a_{ik} = 0$, és ha $k > j$, akkor $b_{kj} = 0$. Így $a_{ik}b_{kj} = 0$, kivéve esetleg az $i \leq k + 1$ és $k \leq j$ eseteket. Ha $i > j + 1$, akkor nincs ilyen k , tehát $c_{ij} = 0$. A másik esetet hasonlóan bizonyíthatjuk. ♦

2.11. Állítás. Legyenek $A, X, H \in \mathbb{C}^{n \times n}$ mátrixok, és legyen $X^{-1}AX = H$ az A mátrix olyan hasonlósági transzformáltja, ami felső Hessenberg-alakú. Továbbá legyen $X = QR$, ahol $Q^H Q = I$, azaz írjuk fel X QR-felbontását (lásd később a 3.2. alfejezetben). Ekkor $Q^H A Q$ szintén felső Hessenberg-mátrix (lásd [9]).

Bizonyítás: $H = R^{-1}Q^H A Q R$ -ből következik, hogy $Q^H A Q = R H R^{-1}$. Mivel R felső háromszög mátrix, az előző állítás alapján könnyű belátni, hogy $R H R^{-1}$ is felső Hessenberg-mátrix. ♦

12. Definíció. Egy n -edrendű felső Hessenberg-mátrix redukálatlan, ha $a_{i+1,i} \neq 0$, $i = 1, 2, \dots, n - 1$. Azaz egy redukálatlan felső Hessenberg-mátrix átló alatti elemei nemnulla elemek.

3. fejezet

Az LR - és QR -algorithmus

3.1. Az LR -algorithmus

Egy $n \times n$ -es $A := [a_{ij}]$ mátrix $A = LU$ (trianguláris) felbontásán az

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ l_{21} & 1 & 0 & \dots & 0 \\ l_{31} & l_{32} & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ l_{n1} & l_{n2} & l_{n3} & \dots & 1 \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & u_{13} & \dots & u_{1n} \\ 0 & u_{22} & u_{23} & \dots & u_{2n} \\ 0 & 0 & u_{33} & \dots & u_{3n} \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & u_{nn} \end{bmatrix}$$

alakú előállításról értjük, ahol L olyan alsó háromszög mátrix, melynek diagonálisában csupa 1-esek állnak, U pedig felső háromszög mátrix. Csak akkor létezik ez a felbontás, ha az A mátrix bal felső aldeteminánsai (főminorai) mind 0-tól különbözőek.

Az LR -algorithmust Rutishauser vezette be 1954-ben. Az LR jelölés a trianguláris felbontás két tényezőjére utal (L : alsó, R : felső háromszög alakú), ahol az L a német „links”, míg az R a „rechts” szavak kezdőbetűje.

Az LR -algorithmus:

```
 $A_1 := A$   
for  $i = 1, 2, 3, \dots$   
     $LU$ -felbontás:  $A_i = L_i R_i$   
     $A_{i+1} = R_i L_i$   
end for  $i$ 
```

feltéve, hogy az $A_i = L_i R_i$ LU -felbontás létezik $\forall i$ -re. Ahogy az előbb említettük, ennek egyik elégséges feltétele az, hogy az A_i mátrixok bal felső aldeteminánsai mind 0-tól különbözőek.

Ez az algoritmus azon a megfigyelésen alapul, hogy az $A_1 = L_1 R_1$ és $A_2 = R_1 L_1 = L_2 R_2$ mátrixok hasonlóak. Ugyanis, L_1 nemszinguláris volta miatt (mert $\det L_1 = 1$) az A_2 mátrix felírható a következő alakban: $R_1 = L_1^{-1} A_1 \Rightarrow A_2 = L_1^{-1} A_1 L_1$. Ebből következik, hogy A_i hasonló A -hoz $\forall i = 1, 2, 3, \dots$ -ra, ugyanis

$$A_{i+1} = R_i L_i = L_i^{-1} A_i L_i = L_i^{-1} L_{i-1}^{-1} A_{i-1} L_{i-1} L_i = \dots = \mathcal{L}_i^{-1} A_1 \mathcal{L}_i,$$

ahol

$$\mathcal{L}_i = L_1 L_2 \dots L_i$$

$$\mathcal{R}_i = R_k R_{k-1} \dots R_1. \quad (3.1)$$

A 2. fejezetben pedig láthattuk, hogy hasonló mátrixok sajátértékei megegyeznek.

3.1. Tétel.

$$A^i = \mathcal{L}_i \mathcal{R}_i \quad i = 1, 2, \dots \quad (3.2)$$

Bizonyítás:

$$\begin{aligned} \mathcal{L}_i \mathcal{R}_i &= \mathcal{L}_{i-1} \underbrace{L_i R_i}_{\mathcal{R}_{i-1}} \mathcal{R}_{i-1} = \mathcal{L}_{i-1} A_i \mathcal{R}_{i-1} = \\ &= \mathcal{L}_{i-2} \underbrace{L_{i-1} R_{i-1} L_{i-1} R_{i-1}}_{\mathcal{R}_{i-2}} \mathcal{R}_{i-2} = \mathcal{L}_{i-2} A_{i-1}^2 \mathcal{R}_{i-2} = \dots = A_1^i = A^i. \blacklozenge \end{aligned}$$

Jelölje $\lambda_1, \lambda_2, \dots, \lambda_n$ az $n \times n$ -es A mátrix sajátértékeit, és x_1, x_2, \dots, x_n a megfelelő sajátvektorokat. Ekkor a kanonikus alak:

$$AX = X\Lambda, \quad (3.3)$$

ahol $X = [x_1, x_2, \dots, x_n]$ és $\Lambda = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_n)$. Ha a sajátvektorok lineárisan függetlenek, akkor a belőlük képzett X mátrix nemszinguláris és (3.3)-ból A kifejezhető:

$$A = X\Lambda X^{-1}. \quad (3.4)$$

3.2. Tétel. *Ha az A mátrix sajátértékei abszolút értékben különbözőek, azaz*

$$|\lambda_1| > |\lambda_2| > \dots > |\lambda_n| > 0,$$

($\Rightarrow A$ nemszinguláris mátrix, és X szintén nemszinguláris), továbbá ha X -nek és $Y := X^{-1}$ -nek létezik LU-felbontása (azaz a bal felső főminorok 0-tól különbözőek), akkor az LR-algoritmus konvergens, és az (A_i) sorozat határértéke egy felső háromszög mátrix (melynek átlójában az A mátrix sajátértékei állnak).

Bizonyítás: [3] alapján.

Tekintsük az X és Y mátrixok trianguláris felbontását, amik a feltétel miatt léteznek: $X = L_X R_X$ és $Y = L_Y R_Y$. Ekkor az A^i hatványt (3.2) máshogy is kiszámolhatjuk - (3.4) alapján:

$$A^i = (X \Lambda X^{-1})^i = X \Lambda^i Y = L_X R_X \Lambda^i L_Y R_Y = L_X R_X \underbrace{\Lambda^i L_Y \Lambda^{-i}} \Lambda^i U_Y = \blacktriangledown.$$

Legyen $L_Y = [t_{jk}]$, ahol $t_{jj} = 1$ és $t_{jk} = 0$, ha $j < k$. Ekkor

$$\Lambda^i L_Y \Lambda^{-i} = \left[\begin{array}{c} \left(\frac{\lambda_j}{\lambda_k} \right)^i t_{jk} \end{array} \right].$$

A sajátértékekere tett feltevés miatt

$$\lim_{i \rightarrow \infty} \Lambda^i L_Y \Lambda^{-i} = I.$$

Ha tehát $E_i := \Lambda^i L_Y \Lambda^{-i} - I$, akkor

$$\lim_{i \rightarrow \infty} E_i = \mathcal{O}.$$

Ezek alapján A i -edik hatványát így írhatjuk tovább:

$$\blacktriangledown = L_X R_X (I + E_i) \Lambda^i R_Y = L_X (I + \underbrace{R_X E_i R_X^{-1}}) R_X \Lambda^i R_Y = L_X (I + F_i) R_X \Lambda^i R_Y = \blacktriangle,$$

ahol

$$\lim_{i \rightarrow \infty} F_i = \mathcal{O}.$$

Tekintsük $I + F_i$ LU -felbontását (létezik, ha i elég nagy): $I + F_i = L^{(i)} R^{(i)}$. Ekkor nyilvánvaló, hogy

$$\lim_{i \rightarrow \infty} L^{(i)} = \lim_{i \rightarrow \infty} R^{(i)} = I,$$

így folytatva az i -edik hatvány felírását:

$$\blacktriangle = \underbrace{L_X L^{(i)}} \underbrace{R^{(i)} R_X \Lambda^i R_Y} = \mathcal{L}_i \mathcal{R}_i,$$

ahol \mathcal{L}_i -t és \mathcal{R}_i -t lásd fentebb (3.1), ugyanis ezt kapjuk, ha összehasonlítjuk az A^i korábbi (3.2) felírásával. Ebből következik, hogy

$$\lim_{i \rightarrow \infty} \mathcal{L}_i = L_X,$$

majd felhasználva, hogy

$$A_{i+1} = \mathcal{L}_i^{-1} A_1 \mathcal{L}_i$$

kapjuk:

$$\lim_{i \rightarrow \infty} A_{i+1} = L_X^{-1} A_1 L_X.$$

Végül $L_X^{-1}X = R_X$ és (3.4) felhasználásával kapjuk, hogy

$$\lim_{i \rightarrow \infty} A_i = L_X^{-1} \overbrace{X \Lambda X^{-1}} L_X = R_X \Lambda R_X^{-1}.$$

Láthatjuk, hogy az eredmény tényleg egy felső háromszög mátrix. ♦

A konvergencia sebessége a

$$q := \max_{1 \leq k < j \leq n} \left| \frac{\lambda_j}{\lambda_k} \right|$$

mennyiségtől függ: minél kisebb a q értéke, annál gyorsabb a konvergencia. A konvergencia gyorsításának érdekében az LR -algoritmust gyakran úgy módosítják, hogy az A_{i+1} mátrixra való áttérés előtt egy $A_i - b_i I$ eltolást alkalmaznak, ahol b_i egy alkalmasan választott szám, csökkentve ezzel a q értékét.

Mivel az LU -felbontás nem mindig létezik, az LR -algoritmus bármelyik lépésben elakadhat. Ezen a problémán úgy lehet segíteni, hogy a felbontás során részleges főelemkiválasztást végzünk. Az így módosított trianguláris felbontás létezésének elégséges feltétele, hogy a szóban forgó mátrix reguláris. Így tehát a *módosított LR -algoritmus* egyetlen lépésben sem akadhat el, ha nonszinguláris A mátrixból indulunk ki. A módosított LR -algoritmusról bővebben olvashatunk a [3] könyvben.

3.2. A QR -algoritmus

Egy A mátrix QR -felbontásán (ortogonális triangularizációján) olyan $A = QR$ felbontást értünk, amikor a trianguláris felbontás L alsó háromszög mátrixát egy Q ortogonális mátrixra cseréljük, viszont az R felső háromszög mátrixot továbbra is megtartjuk második tényezőként.

3.3. Tétel. *Tetszőleges $n \times n$ -es A mátrix előállítható egy Q ortogonális és egy R felső háromszög mátrix szorzataként: $A = QR$.*

Emlékeztető: Az $A = LU$ -felbontás nem biztos, hogy létezik, még abban a speciális esetben sem, ha az A mátrix nonszinguláris. Például a következő mátrix nonszinguláris, még sincs LU -felbontása:

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

Az LU -felbontást, ha létezik, úgy tesszük egyértelművé, hogy az L mátrix diagonálisában az elemeket 1-eseknek választjuk (mi már így definiáltuk a felbontást).

3.4. Tétel. *Ha az A mátrix nonszinguláris, akkor az $A = QR$ felbontás a Q oszlopainak és az R sorainak előjelétől eltekintve egyértelmű.*

Például:

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = \begin{bmatrix} q_{11} & q_{12} \\ q_{21} & q_{22} \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} \\ 0 & r_{22} \end{bmatrix} = \begin{bmatrix} -q_{11} & q_{12} \\ -q_{21} & q_{22} \end{bmatrix} \begin{bmatrix} -r_{11} & -r_{12} \\ 0 & r_{22} \end{bmatrix}.$$

Állapodjunk meg abban, hogy a továbbiakban az R mátrix diagonális elemeit pozitívnak választjuk. Ekkor már a felbontás egyértelmű.

Bizonyítás: [3] alapján.

Tegyük fel, hogy $A = Q_1 R_1 = Q_2 R_2$. Mivel a feltétel szerint A nemszinguláris, ezért $0 \neq \det A = \det Q \det R$, amiből következik, hogy $\det R \neq 0$ (definíció szerint $Q^T = Q^{-1}$, így $\det Q \neq 0$), vagyis U sem szinguláris. Átrendezéssel kapjuk, hogy

$$Q_2^T Q_1 = R_2 R_1^{-1} =: B.$$

Tudjuk, hogy felső háromszög mátrixok inverze felső háromszög mátrix, és felső háromszög mátrixok szorzata is felső háromszög mátrix (bizonyítása könnyű, hasonló a 2.10. állítás bizonyításához), ezért B felső trianguláris mátrix. Másrészt $B^T B = Q_1^T Q_2 Q_2^T Q_1 = I$, azaz $B^T = B^{-1}$. Ebből következik, hogy B csak diagonális mátrix lehet, azaz $B = \text{diag}(b_{ii})$ alakú, ahol $b_{ii} = (b_{ii})^{-1}$, $i = 1, 2, \dots, n$. Így $(b_{ii})^2 = 1$, $b_{ii} = \pm 1$. A fentiek alapján pedig $B^{-1} = B$, $Q_1 = Q_2 B$ és $R_1 = B R_2$. ♦

A QR -algoritmust Francis vezette be 1961-ben. Az elnevezésben szereplő Q és R betűk az ortogonális triangularizáció két tényezőjére utalnak (Q ortogonális, R felső háromszög mátrix). A sajátprobléma megoldására ez az egyik legjobb módszer.

A QR -algoritmus a következő rekurzióval adható meg:

```

A1 := A
for i = 1, 2, 3, ...
    ortogonális felbontás: Ai = QiRi
    Ai+1 = RiQi
end for i

```

Ha A nemszinguláris és R_i átlós elemeit pozitívnak választjuk, akkor a 3.4. tétel szerint a Q_i és R_i tényezők egyértelműen meghatározottak. Az algoritmus azon az észrevételre alapul, hogy az $A_1 = Q_1 R_1$ és $A_2 = R_1 Q_1 = Q_2 R_2$ mátrixok hasonlóak, ugyanis $Q^{-1} = Q^T$, így $R_1 = Q_1^T A_1 \Rightarrow A_2 = Q_1^T A_1 Q_1$, és fentebb már láttuk, hogy hasonló mátrixok sajátértékei megegyeznek.

A QR -algoritmus különösen akkor ajánlott, mikor a sajátértékek legalább negyede keresett, pl. az első $n/4$ sajátértéket szeretnénk meghatározni.

3.5. Állítás. *A QR-algoritmus megőrzi a kiindulási A mátrix felső Hessenberg-alakját.*

Bizonyítás: Ha $A_i = Q_i R_i$ felső Hessenberg-mátrix, akkor - mivel R_i felső háromszög mátrix - a 2.10. állítás alapján Q_i felső Hessenberg-mátrix. Ekkor $A_{i+1} = R_i Q_i$ szintén felső Hessenberg-mátrix lesz, azaz a QR-algoritmus minden lépésében megmarad a felső Hessenberg-alak, ha abból indultunk ki. ♦

Amennyiben a mátrix sajátértékei különböző abszolút értékűek, és a bal oldali sajátvektorok mátrixának van LU-felbontása, akkor az LR-algoritmus bizonyításához hasonlóan belátható, hogy a QR-algoritmus egy felső háromszög mátrixhoz konvergál (ld. [3]).

4. fejezet

A QR -algorithmus változatai

Ennek a fejezetnek az ismertetését nagy mértékben Stewart [1] nyomán tesszük.

4.1. Az explicit eltolásos QR -algorithmus

Vizsgáljuk meg a QR -algorithmust. Már tudjuk, hogy az A mátrixot unitér (valós esetben ortogonális) hasonlósági transzformációkkal alakítja át a következőképpen:

$$A_{i+1} = Q_i^H A_i Q_i, \quad i = 1, 2, 3, \dots, \quad \text{ha} \quad A_1 = A.$$

Most csak egy lépést fogunk ebből megnézni, ezért elhagyhatjuk az i alsó indexet.

A QR -algorithmus egy iteratív módszer a Schur-normálformára való redukálásra, ezért célunk, hogy az utolsó sor első $n-1$ elemét minél kisebbé tegyük. Írjuk fel A -t a következő alakban:

$$\mathbf{A} = \begin{pmatrix} B & h \\ g^H & \mu \end{pmatrix}. \quad (4.1)$$

Tehát az a célunk, hogy $\|g^H\|_2 \rightarrow 0$. Ehhez Q -t úgy próbáljuk megválasztani, hogy redukáljuk ezt a normát. Amint g^H lényegében nullává válik (azaz megvan a sajátérték), folytathatjuk az eljárást a kisebb B mátrixon (ez az eljárás a *defláció*). Legyen q a λ sajátértékhez tartozó bal sajátvektor. Vegyük észre, hogy ha a q -t így választjuk, akkor a

$$\mathbf{Q} = (Q_* \quad q),$$

tökéletes lenne. Ebben az esetben ugyanis (Schur-tétel)

$$Q^H A Q = \begin{pmatrix} Q_*^H A Q_* & Q_*^H A q \\ 0 & \lambda \end{pmatrix}.$$

Vagyis azonnal továbbléphetnénk a $Q_*^H A Q_*$ redukálására.

Pontos sajátvektor hiányában megpróbálkozhatunk egy közelítő sajátvektor kiszámításával, amivel Q -t definiálhatjuk. Ha g^H kicsi, az e_n^T vektor látszik célszerűnek az A egy bal sajátvektorának közelítésére. Ekkor $q = e_n$ választással

$$Q = \begin{pmatrix} Q_* & 0 \\ 0 & 1 \end{pmatrix}$$

alakú lenne, ahol Q_* unitér (mert Q -nak unitérnek kell lennie). Ez azonban nem jó, mert a $Q^H A Q$ transzformáció a g^H -t $g^H Q_*$ -ba vinné, aminek a normája változatlan: $\|g^H Q_*\|_2 = \|g^H\|_2$ (lásd 2.2. állítás). Ehelyett q^H -t az inverz iteráció segítségével fogjuk meghatározni. Az eltolást egy egyelőre meg nem határozott κ -val fogjuk végezni. Oldjuk meg q^H -ra a következőt (mivel egy bal sajátvektort próbálunk közelíteni):

1. $q^H(A - \kappa I) = e_n^T$
2. $\hat{q} = q/\|q\|_2$

Így az alábbi formulát kapjuk a q^H vektorra:

$$q^H = \frac{e_n^T(A - \kappa I)^{-1}}{\|e_n^T(A - \kappa I)^{-1}\|_2}. \quad (4.2)$$

Belátjuk, hogy a QR -algoritmus éppen a fentieknek megfelelő $(Q_* \quad q)$ mátrixot használja. Írjuk fel az eltoló mátrix QR -felbontását:

$$A - \kappa I = QR.$$

Ebből

$$Q^H = R(A - \kappa I)^{-1}.$$

Tudjuk, hogy $e_n^T R = r_{nn} e_n^T$. Ezek alapján pedig kapjuk, hogy:

$$q^H \equiv e_n^T Q^H = r_{nn} e_n^T (A - \kappa I)^{-1}$$

éppen a keresett (4.2) vektor ($\|q\|_2 = 1$), és ez éppen a κ -hoz legközelebbi sajátértékhez tart (itt megvalósul az inverz iteráció egy lépése). Ezzel meghatároztunk egy Q -t (tükrözéssel mindig felírható egy olyan Q , aminek az utolsó oszlopa q). Az eltoló mátrix QR -felbontásából következik, hogy

$$RQ = Q^H(A - \kappa I)Q = Q^H A Q - \kappa I.$$

Tehát a hasonlósági transzformációt így tudjuk elvégezni:

$$Q^H A Q = RQ + \kappa I.$$

Ezek alapján meg is van az algoritmus.

Az explicit eltolásos QR -algorithmus:

```
 $A_1 := A$   
for  $i = 1, 2, 3, \dots$   
    válasszunk egy  $\kappa_i$  eltolást  
     $A_i - \kappa_i I = Q_i R_i$ , ahol  $Q_i R_i$  a  $QR$ -felbontás  
     $A_{i+1} = R_i Q_i + \kappa_i I$   
end for  $i$ 
```

A fentiek szerint pedig látjuk, hogy az eltolásos algoritmusnál is igaz az a tény, hogy az A_1 és A_2 mátrixok hasonlóak. Ugyanis:

$$A_1 - \kappa I = Q_1 R_1 \quad \Rightarrow \quad Q_1^H (A_1 - \kappa I) = R_1 \quad \Rightarrow$$
$$A_2 = R_1 Q_1 + \kappa I = Q_1^H (A_1 - \kappa I) Q_1 + \kappa I = Q_1^H A_1 Q_1$$

4.1. Lemma. *Ha a κ eltolás megegyezik egy egyszeres sajátértékkel, azaz $\kappa = \lambda$, akkor a QR -algorithmus egy lépéséből meghatározhatóak a λ -hoz tartozó bal és jobb oldali sajátvektorok.*

Bizonyítás: Mivel az A mátrix $\mathbb{R}^{n \times n}$ -es (most valósban bizonyítunk), az $A - \kappa I$ rangja $n - 1$. Ha a QR -felbontást Householder-tükrözésekkel készítjük, akkor $Q \in \mathbb{R}^{n \times n}$ ortogonális mátrix, így R rangja $n - 1$ lesz. Tegyük fel, hogy R utolsó átlóeleme zérus, azaz $r_{nn} = 0$. Ekkor y^T bal oldali sajátvektor, ha $\underbrace{y^T Q}_R = 0$. Könnyen ellenőrizhető, hogy R bal sajátvektora e_n^T , így a bal sajátvektorra igaz:

$$y^T Q = e_n^T \quad \rightsquigarrow \quad e_n^T Q^T = y^T.$$

A jobb oldali sajátvektorhoz particionáljuk R -et és annak megfelelően x -et is:

$$\begin{pmatrix} R_{11} & h \\ 0 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = 0.$$

Itt vegyük észre, hogy ha a jobb sajátvektor utolsó eleme nulla lenne, akkor R_{11} invertálhatósága miatt $x_1 = 0$ következne. Emiatt az utolsó elemet 1-nek választhatjuk. Kifejezve x_1 -et az egyenletből, kapjuk:

$$R_{11} x_1 = -h \quad \rightsquigarrow \quad x_1 = -R_{11}^{-1} h.$$

Ha a QR -felbontás során A -nak valamely közbülső oszlopa lineárisan összefüggőnek adódna, akkor ezt az utolsó helyre mozdíthatjuk. De könnyen meggyőződhetünk róla, hogy ez érdemben nem befolyásolja a fenti gondolatmenetünket. \blacklozenge

Most vizsgáljuk meg az algoritmus konvergencia tulajdonságait. Globális konvergenciát csak nagyon speciális esetekben tudunk megállapítani, így most csak lokális konvergencia bizonyítékot adunk.

Stewart felismerése alapján bevezethetjük az alábbi tételt:

4.2. Tétel. *A QR-algoritmus végrehajtása során a k -adik lépésben legyen $q_k^H = e_n^T Q_k^H$. Ekkor fix κ eltolás esetén*

$$q_k^H = e_n^T (A_1 - \kappa I)^{-k} Q_1 Q_2 \cdots Q_{k-1}, \quad (4.3)$$

ami a hasonlósági transzformált mátrixra nézve az inverz iteráció k -adik sajátvektor közelítése (itt az egyenlőség csak irányban értendő).

Bizonyítás: Tudjuk, hogy $q_1^H = e_n^T (A_1 - \kappa I)^{-1}$. Használjuk fel, hogy $A_2 = Q_1^H A_1 Q_1$. Ekkor a q_2^H -t így írhatjuk fel:

$$q_2^H = e_n^T (A_2 - \kappa I)^{-1} = \underbrace{e_n^T Q_1^H}_{q_1^H} (A_1 - \kappa I)^{-1} Q_1 = \underbrace{q_1^H (A_1 - \kappa I)^{-1} Q_1}_{q_2^H} = e_n^T (A_1 - \kappa I)^{-2} Q_1.$$

Most alkalmazzunk A -ra egy hasonlósági transzformációt: $X^{-1}AX$. Tudjuk, hogy ha v^T az A egy bal oldali sajátvektora, akkor a hasonlósági transzformáció során ez így változik meg: $v^T X$ (lásd a 9. definíció után). Ez alapján az előbbi eredményünk az $A_2 = Q_1^H A_1 Q_1$ mátrixszal készített inverz iteráció vektora.

Most nézzük a k -adik esetet. Tudjuk, hogy $q_k^H = e_n^T (A_k - \kappa I)^{-1}$. Ezt így lehet tovább írni a fentiek alapján:

$$\begin{aligned} q_k^H &= e_n^T Q_{k-1}^H (A_{k-1} - \kappa I)^{-1} Q_{k-1} = q_{k-1}^H (A_{k-1} - \kappa I)^{-1} Q_{k-1} = \\ &= e_n^T (A_{k-1} - \kappa I)^{-2} Q_{k-1} = \dots = e_n^T (A_1 - \kappa I)^{-k} Q_1 Q_2 \dots Q_{k-1}. \end{aligned}$$

Ez pedig éppen az A_1 mátrixszal végzett inverz iteráció k -adik vektora a hasonlósági transzformáció után. ♦

1. Következmény. *Abszolút értékben egyszeres gyökök esetén az inverz iteráció alapján a QR-algoritmus konvergens lesz.*

3. Megjegyzés. *Ha a κ eltolás mindig ugyanaz, akkor a konvergencia-sebesség csak lineáris.*

A továbbiakban az algoritmus egy lépését vizsgáljuk, ezért hagyjuk el megint az alsó indexet.

Az A mátrix fentebbi alakja alapján a következő particionált alakot írhatjuk le:

$$A - \kappa I \equiv \begin{pmatrix} B - \kappa I & h \\ g^H & \mu - \kappa \end{pmatrix} = \begin{pmatrix} P & f \\ e^H & \pi \end{pmatrix} \begin{pmatrix} S & \rho \\ 0 & r \end{pmatrix} \equiv QR. \quad (4.4)$$

Írjuk fel az algoritmus soron következő mátrixát is: $\hat{A} = RQ + \kappa I$. Ekkor az $\hat{A} - \kappa I = RQ$ particionált alakja:

$$\hat{A} - \kappa I \equiv \begin{pmatrix} \hat{B} - \kappa I & \hat{h} \\ \hat{g}^H & \hat{\mu} - \kappa \end{pmatrix} = \begin{pmatrix} S & \rho \\ 0 & r \end{pmatrix} \begin{pmatrix} P & f \\ e^H & \pi \end{pmatrix} \equiv RQ. \quad (4.5)$$

Egy korlátot szeretnénk adni $\|\hat{g}\|_2$ -ra a $\|g\|_2$ -val kifejezve, méghozzá úgy, hogy ez egy felső korlát legyen.

Mivel Q unitér, az utolsó sorának és oszlopának a normája egy kell, hogy legyen. Így

$$\|e\|_2^2 + \pi^2 = \|f\|_2^2 + \pi^2 = 1 \quad \Rightarrow \quad \|e\|_2 = \|f\|_2. \quad (4.6)$$

A következő lépésben azt szeretnénk megmutatni, hogy e kicsi, ha g kicsi. Az első particionált formából (4.4) számolva $g^H = e^H S$. Ha feltételezzük, hogy S nonsinguláris és azt mondjuk, hogy $\|S^{-1}\|_2 = \sigma$, akkor

$$e^H = g^H S^{-1} \quad \Rightarrow \quad \|e\|_2 \leq \sigma \|g\|_2. \quad (4.7)$$

Szükségünk van r egy korlátjára is. Ha kihasználjuk, hogy Q unitér, azaz az első particionált alakot balról szorozzuk Q^H -val, akkor a következő alakot kapjuk:

$$\begin{pmatrix} P^H & e \\ f^H & \pi \end{pmatrix} \begin{pmatrix} B - \kappa I & h \\ g^H & \mu - \kappa \end{pmatrix} = \begin{pmatrix} S & \rho \\ 0 & r \end{pmatrix},$$

ebből pedig $r = f^H h + \pi(\mu - \kappa)$. (4.6)-ból és (4.7)-ből, valamint abból a tényből, hogy $\pi^2 \leq 1 \Rightarrow |\pi| \leq 1$, következik, hogy

$$\|r\|_2 \leq \sigma \|g\|_2 \|h\|_2 + |\mu - \kappa|. \quad (4.8)$$

A második particionált felírásból (4.5) számolva $\hat{g}^H = r e^H$. A két felső korlátot - (4.7) és (4.8) - felhasználva

$$\|\hat{g}\|_2 \leq \sigma^2 \|h\|_2 \|g\|_2^2 + \sigma |\mu - \kappa| \|g\|_2$$

adódik. Visszaírva az alsó indexeket, kapjuk:

$$\|g_{i+1}\|_2 \leq \sigma_i^2 \|h_i\|_2 \|g_i\|_2^2 + \sigma_i |\mu_i - \kappa_i| \|g_i\|_2.$$

Ez a felső korlát az sugallja, hogy $\kappa_i = \mu_i$ -t kellene választanunk (ahol $\mu_i = a_{nn}$), hogy a jobb oldal második tagját nullával tegyük egyenlővé. Ekkor ezt kapjuk:

$$\|g_{i+1}\|_2 \leq \sigma_i^2 \|h_i\|_2 \|g_i\|_2^2. \quad (4.9)$$

Ezt a $\kappa = \mu$ eltolást *Rayleigh eltolásnak* nevezzük, mert μ az e_n vektorhoz tartozó Rayleigh-hányados. Ugyanis

$$e_n^T \begin{pmatrix} & & & \\ & & & \\ & & & \\ 0 & \dots & 0 & | a_{nn} \end{pmatrix} = a_{nn} e_n^T,$$

ahol e_n^T közelítő sajátvektor. Ekkor pedig $x = e_n$ választással a Rayleigh-hányados: $\frac{x^T A x}{x^T x} = a_{nn} = \mu$, ami nem más, mint a sajátérték közelítése.

A módszer konvergenciájának vizsgálatához tegyük fel, hogy

$$\sigma \geq \|S_i^{-1}\|_2 \quad \text{és} \quad \eta \geq \|h_i\|_2.$$

A második feltevés mindig teljesül $\eta = \|A\|_2$ -ra, az első pedig ésszerű, ha egy egyszeres λ sajátértékhez konvergálunk. Ekkor a κ_i eltolások λ -hoz közelítenek és S_i az $A - \lambda I$ R -tényezőjének fő almatrixához konvergál. Továbbá, mivel λ egyszeres, S_i inverzére adunk felső korlátot. Ezek alapján a következőképpen írhatjuk át (4.9)-et :

$$\|g_{i+1}\|_2 \leq \eta \sigma^2 \|g_i\|_2^2.$$

Most tegyük fel, hogy eljutottunk addig a pontig, ahol

$$\eta \sigma^2 \|g_i\|_2 < 1. \tag{4.10}$$

Ekkor még tovább becsülhetjük (4.9)-et:

$$\|g_{i+1}\|_2 \leq (\eta \sigma^2 \|g_i\|_2) \|g_i\|_2 < \|g_i\|_2.$$

A következő iterációnál ezt kapjuk:

$$\|g_{i+2}\|_2 \leq (\eta \sigma^2 \|g_i\|_2)^2 \|g_i\|_2,$$

és általánosságban vége:

$$\|g_{i+j}\|_2 \leq (\eta \sigma^2 \|g_i\|_2)^j \|g_i\|_2.$$

Mivel g_i nullához tart, az explicit eltolásos QR -algoritmus konvergens.

A fenti levezetésből következik, hogy az explicit eltolásos QR -algoritmus valós, egyszeres sajátérték esetén konvergens, ha abszolút értékben egy sajátérték van az eltolási paraméterhez a legközelebb. (4.10) teljesülése esetén érkezünk el arra a pontra, amikor rátérhetünk a Rayleigh eltolásra. Ekkor már kvadratikusan a konvergencia.

Itt megjegyezzük, hogy komplex konjugált gyökpár esetén (mivel ilyenkor az inverz iteráció oszcillál) külön eljárást kell alkalmazni, amit később ismertetünk.

A konvergencia ismétlődések kielégítik a következőt:

$$\|g_{i+1}\| \leq (\eta\sigma^2)\|g_i\|_2^2.$$

Az ilyen jellegű konvergenciát kvadrátikusnak nevezzük, és ez nagyon gyors. Ha például $\eta\sigma^2 = 1$ és $\|g_0\|_2 = 0, 1$, akkor a soron következő iterációkra a felső korlátok:

$$\|g_1\|_2 \leq 10^{-2}, \|g_2\|_2 \leq 10^{-4}, \|g_3\|_2 \leq 10^{-8}, \|g_4\|_2 \leq 10^{-16}, \dots$$

4. Megjegyzés. Ha A hermitikus, azaz $A = A^H$, akkor a konvergencia sebesség még nagyobb lesz.

Ebben az esetben $g_i = h_i$, és η -t vehetjük $\eta = \|g_i\|_2$ -nak, így

$$\|g_{i+1}\| \leq \sigma^2\|g_i\|_2^3.$$

Ha $\sigma = 1$ és $\|g_0\| = 0, 1$, akkor az iterációk felső korlátai: $10^{-3}, 10^{-9}, 10^{-27}, \dots$. Az ilyen fajta konvergenciát köbös vagy harmadrendű konvergencia sebességnek nevezzük.

A levezetések alapján kimondhatjuk az alábbi tételt:

4.3. Tétel. Egyszeres sajátérték esetén a Rayleigh eltolásos QR -algorithmus konvergencia sebessége másodrendű, és ha a mátrix még hermitikus is, akkor harmadrendű.

Az algoritmus kitűnő konvergencia tulajdonságokkal rendelkezik. Azonban problémát jelent, hogy megköveteli az A_i mátrixok QR -felbontásának kiszámítását. Ez teljes mátrixok esetén $O(n^4)$ műveletigényű algoritmushoz vezet. (Mindig fontos, hogy egy algoritmus stabil, megbízható és minél gyorsabb legyen.) Egy másik probléma a valós mátrixokkal kapcsolatos. Mivel az explicit eltolásos QR -algoritmus egy κ eltolással ellátott inverz iteráció alkalmazása, elkerülhetetlen, hogy κ a keresett sajátértékhez közel essen. Ha azonban ez a sajátérték komplex, akkor komplex eltolást kell alkalmaznunk, ekkor pedig A_i komplex mátrix lesz és az algoritmus alkalmazása több, mint kétszer annyi erőfeszítéssel fog járni. Ráadásul a kerekítési hibák miatt a sajátértékek nem pontos konjugált párokban fognak megjelenni.

5. Megjegyzés. Az első problémára megoldást jelenthet, hogy a mátrixot az algoritmus elvégzése előtt felső Hessenberg-alakra hozzuk, amelyet könnyű tényezőkre bontani és ez az alak végig változatlan marad (3.5. állítás). A második problémát a valós Schur forma kiszámításával küszöbölhetjük ki.

4.2. QR -algoritmus Hessenberg-mátrixokra

Ha el tudjuk fogadni a komplex aritmetikát, megfelelő algoritmus áll a rendelkezésünkre egy általános mátrix Schur normálformájának kiszámítására. Először is redukáljuk a mátrixot Hessenberg-alakúra, majd alkalmazzuk a QR -algoritmust Rayleigh eltolásokkal. Tipikus esetben az algoritmus csak néhány lépés után kezd el konvergálni. Amint $a_{n,n-1}$ elég kicsi számmá válik, nullának vesszük, és folytathatjuk úgy az algoritmust, hogy $a_{n-1,n-1}$ -et használjuk eltolásként (deflációs lépéssel eggyel csökkentjük a mátrix méretét). Miközben a jelenlegi eltolás melletti szubdiagonális elem kvadratikusan konvergál a nullához, a többi átló alatti elem is szép lassan a nullához tart. (Ez még az eltolás nélküli QR -algoritmus és a hatványmódszer közötti kapcsolatból következik.) Következésképpen, a deflációk során minél feljebb tartunk a mátrixban, annál kevesebb iteráció szükséges egy sajátérték megtalálásához.

Tegyük fel, hogy a mátrixot már a következő formára redukáltuk:

$$\left(\begin{array}{cccccccc} a & a & a & a & a & a & a & a \\ a & a & a & a & a & a & a & a \\ 0 & a & a & a & a & a & a & a \\ 0 & 0 & a & a & a & a & a & a \\ 0 & 0 & 0 & a & a & a & a & a \\ \hline 0 & 0 & 0 & 0 & 0 & a & a & a \\ 0 & 0 & 0 & 0 & 0 & 0 & a & a \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & a \end{array} \right),$$

így már három sajátértéket megtaláltunk és még öt kell. A következő QR lépés négy transzformációt határoz meg. Az, hogy hogyan alkalmazzuk ezeket a mátrixra, attól függ, hogy a Schur normálformát (2.7. tétel) szeretnénk kiszámítani, vagy csak A sajátértékeit akarjuk meghatározni. Ha egy Schur normálformát szeretnénk kiszámítani, akkor az egész mátrixon alkalmaznunk kell a transzformációkat - vagy legalábbis a vonal fölötti elemeken. Ha azonban csak a sajátértékekre vagyunk kíváncsiak, akkor csak a legelső, legfontosabb almatrixszal kell dolgoznunk:

$$\left(\begin{array}{cccc|ccc} a & a & a & a & a & a & a \\ a & a & a & a & a & a & a \\ 0 & a & a & a & a & a & a \\ 0 & 0 & a & a & a & a & a \\ 0 & 0 & 0 & a & a & a & a \\ \hline 0 & 0 & 0 & 0 & 0 & a & a \\ 0 & 0 & 0 & 0 & 0 & 0 & a \\ 0 & 0 & 0 & 0 & 0 & 0 & a \end{array} \right).$$

A Schur normálformából sajátvektorokat tudunk kiszámítani. Hogy egy λ sajátértékhez a megfelelő sajátvektort ki tudjuk számítani, osszuk fel a Schur formát, hogy felfedjük a sajátértéket:

$$\begin{pmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \lambda \begin{pmatrix} x_1 \\ x_2 \end{pmatrix},$$

ahol λ legyen az A_{11} utolsó diagonáeleme, és A_{22} minden diagonáeleme különbözzön ettől. Ekkor így írható fel a mátrixunk:

$$\begin{pmatrix} B & c & D \\ 0 & \lambda & e^T \\ 0 & 0 & F \end{pmatrix}.$$

Írjuk át az $Ax = \lambda x$ egyenletet így: $(A - \lambda I)x = 0$. Ez az előbbi mátrix alakkal felírva:

$$\begin{pmatrix} B - \lambda I & c & D \\ 0 & \lambda - \lambda & e^T \\ 0 & 0 & F - \lambda I \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = 0, \quad (4.11)$$

mert $F - \lambda I$ invertálható. Alulról felfele haladva az utolsó blokk-egyenletet megoldva kapjuk, hogy $x_3 = 0$. Ezek alapján végül erre az egyenletre jutunk:

$$\begin{pmatrix} B & c & D \\ 0 & \lambda & e^T \\ 0 & 0 & F \end{pmatrix} \begin{pmatrix} w \\ 1 \\ 0 \end{pmatrix} = \lambda \begin{pmatrix} w \\ 1 \\ 0 \end{pmatrix}. \quad (4.12)$$

Ebből $Bw + c = \lambda w$, azaz $(B - \lambda I)w = -c$. Ezt a felső háromszög rendszert meg tudjuk oldani w -re, így az eredeti mátrix sajátvektorát ezek alapján a következő adja meg:

$$x = U \begin{pmatrix} w \\ 1 \\ 0 \end{pmatrix}, \quad (4.13)$$

ahol U a felső Hessenberg-alakra hozásnál használt unitér mátrix. A későbbiekben látni fogjuk, hogyan tudjuk ezt a mátrixot meghatározni.

Ez az eljárás kicsit kockázatosnak tűnhet, ha A -nak többszörös sajátértékei is vannak, mivel ilyenkor λ B átlójában is megjelenhet.

4.3. Az implicit eltolásos QR -algoritmus

Az előbbiekben leírt algoritmus jó módszer egy általános komplex mátrix sajátértékeinek és sajátvektorainak kiszámítására. Valós mátrixok esetén azonban megoldható, hogy a mátrix valós Schur formáját valós aritmetikával számoljuk ki.

Sajátérték-problémák megoldása esetén alapvetően redukálatlan felső Hessenberg-mátrixokkal van dolgunk. Ha egy mátrix redukált, akkor például a következő formát veszi fel:

$$\left(\begin{array}{cc|cccc} * & * & * & * & * & * \\ * & * & * & * & * & * \\ \hline 0 & 0 & * & * & * & * \\ 0 & 0 & * & * & * & * \\ 0 & 0 & 0 & * & * & * \\ 0 & 0 & 0 & 0 & * & * \end{array} \right)$$

Ebben az esetben a probléma két részre oszlik - a fenti ábra szerint egy 2×2 -es és egy 4×4 -es problémára.

Redukálatlan Hessenberg-mátrixok esetén a redukálást érintő legalapvetőbb tény a következő:

4.4. Állítás. *Legyen $A \in \mathbb{R}^{n \times n}$. Ha $Q^T A Q = H$, ahol Q ortogonális és H redukálatlan felső Hessenberg-mátrix, akkor Q első oszlopa egyértelműen meghatározza Q és H mátrixokat.*

Az állítás feltételében pont egy korábbi tételünk van megfogalmazva, miszerint minden mátrix unitér (valóban ortogonális) hasonlósági transzformációval felső Hessenberg-alakra hozható. Ekkor H sajátértékei megegyeznek A sajátértékeivel.

Bizonyítás: Az *Arnoldi-módszerrel* bizonyítunk. Rendezzük át az állításban szereplő egyenletet:

$$A Q = Q H, \quad (4.14)$$

majd pedig írjuk fel a következő formában:

$$A \begin{pmatrix} q_1 & q_2 & q_3 & \dots \end{pmatrix} = \begin{pmatrix} q_1 & q_2 & q_3 & \dots \end{pmatrix} \begin{pmatrix} h_{11} & h_{12} & h_{13} & \dots \\ h_{21} & h_{22} & h_{23} & \dots \\ 0 & h_{32} & h_{33} & \dots \\ \vdots & \vdots & \vdots & \end{pmatrix}. \quad (4.15)$$

Így most q_1 meghatározza h_{11} -et, ugyanis $h_{11} = q_1^T A q_1$. (4.15) első oszlopát kiszámolva azt látjuk, hogy

$$A q_1 = h_{11} q_1 + h_{21} q_2,$$

amiből átrendezéssel a következőt kapjuk:

$$q_2 = \frac{A q_1 - h_{11} q_1}{h_{21}},$$

ahol h_{21} -et a $\|q\|_2 = 1$ feltétel határozza meg (a 2.2. állítás miatt). Mivel H redukálatlan, h_{21} -nek szükségszerűen nemnullának kell lennie. Rendelkezésünkre áll $h_{21} = q_1^T A q_2$ és $h_{22} = q_2^T A q_2$ is (tehát ezek is meghatározhatóak q_1 segítségével), így (4.15) második oszlopa:

$$Aq_2 = h_{12}q_1 + h_{22}q_2 + h_{32}q_3,$$

amiből átrendezéssel

$$q_3 = \frac{Aq_2 - h_{12}q_1 - h_{22}q_2}{h_{32}},$$

ahol h_{32} nemnulla normalizációs állandó. Általánosságban véve

$$h_{i,j-1} = q_i^T A q_{j-1}, \quad i = 1, 2, \dots, j-1, \quad (4.16)$$

és

$$g_j = \frac{Aq_{j-1} - \sum_{i=1}^{j-1} h_{i,j-1}q_i}{h_{j,j-1}}. \quad (4.17)$$

Így q_1 ismeretében ki tudjuk számítani Q -t és H -t is. ♦

Arnoldi-módszer

Ez a *Krilov-bázis* vektorainak - x, Ax, A^2x, \dots , ahol $x \neq 0$ tetszőleges induló vektor - Gram-Schmidt ortogonalizációja. Az Arnoldi-módszernél ennek alapján $q_1 = x/\|x\|_2$ legyen az induló vektor, és tegyük fel, hogy már elkészítettünk $j-1$ db ortonormált vektort, melyeket egy mátrixba rendezünk: $Q = [q_1 \ q_2 \ \dots \ q_{j-1}]$. A következő vektort úgy készítjük, hogy az Aq_{j-1} vektort ortogonalizáljuk a meglévőkre:

$$h_{j,j-1}q_j = (I - QQ^T)Aq_{j-1}, \quad (4.18)$$

ahol $h_{j,j-1}$ a vetítés eredményeként kapott vektor $\|\cdot\|_2$ -ja, mert q_j normált. Kissé átírva az egyenletet kapjuk, hogy

$$AQ = QH + h_{j,j-1}q_j e_{j-1}^T.$$

Q^T -tal való balról szorzás után az egyenletünk $Q^T A Q = H$ lesz, amiből átrendezés után éppen (4.14)-et kapjuk. Láthatjuk, hogy az Arnoldi-módszer is egy olyan ortogonális transzformációt szolgáltat, amivel Hessenberg-alakra hozhatunk.

Térjünk vissza az állításunkhoz. Mivel az Aq_{j-1} -et a korábbi q_i vektorokkal ortogonalizáltuk, az algoritmus az ortogonális polinomok létrehozásának rekurzióját is felidézi.

Azonban ha A szimmetrikus, akkor $Q^T A Q$ is az, és így tridiagonális is. Ekkor a (4.17)-es rekurzió a következőképpen egyszerűsíthető le:

$$q_j = \frac{Aq_{j-1} - h_{j-1,j-1}q_{j-1} - h_{j-2,j-1}q_{j-2}}{h_{j,j-1}}.$$

Az előbbiekben láttuk, hogy az Arnoldi-sémát a q_1 indító vektor egyértelműen meghatározza, így a többi adatot rögzíteni lehet vele. Ez alapján az implicit eltolásos QR -algoritmus lényege az, hogy először meghatározzuk a q_1 vektort, és csak utána hozzuk Hessenberg-alakra a mátrixot.

Az induló vektort az eltolt mátrix első oszlopából vesszük: $q_1 = (A - \kappa I)e_1$, és ezzel a q_1 vektorral készítjük el az első hasonlósági transzformációt. A Householder-tükrözésekben az ortogonális mátrix olyan, hogy első sora e_1^T , első oszlopa e_1 : $Qe_1 = e_1$, $e_1^T Q = e_1^T$. Ez a későbbiekben arra lesz jó, hogy valós transzformációkon keresztül is megvalósítható legyen a dupla eltolásos QR -algoritmus.

4.4. Az implicit dupla eltolásos QR -algoritmus

Most egy olyan megoldást vizsgálunk, amivel valós Schur forma kiszámítása esetén el tudjuk kerülni a komplex aritmetikát, ha A egy valós Hessenberg-mátrix. A módszer lényege az, hogy két komplex konjugált eltolást végzünk el egyszerre.

Az implicit dupla eltolásos QR -algoritmus direkt változata:

```

 $A_1 := A$ 
for  $i = 1, 2, 3, \dots$ 
  1.  $A_i - \kappa I = Q_i R_i$ 
      $A_{i+1} = Q_i^H A_i Q_i$ 
  2.  $A_{i+1} - \bar{\kappa} I = Q_{i+1} R_{i+1}$ 
      $A_{i+2} = Q_{i+1}^H A_{i+1} Q_{i+1}$ 
end for  $i$ 

```

Vizsgáljuk meg ezt az algoritmust. Először is nézzük az alábbi két lépést:

1. $A_1 - \kappa I = Q_1 R_1, \quad A_2 = Q_1^H A_1 Q_1,$
2. $A_2 - \bar{\kappa} I = Q_2 R_2, \quad A_3 = Q_2^H A_2 Q_2.$

Azt állítjuk, hogy $Q = Q_1 Q_2$ esetén $A_3 = Q^T A_1 Q$ valós. Ehhez először is vegyük észre, hogy a

$$(A_1 - \bar{\kappa}I)(A_1 - \kappa I) = A_1^2 - 2\Re(\kappa)A_1 + |\kappa|^2 I$$

mátrix valós. Ugyanakkor azonban az algoritmus felírásának felhasználásával

$$\begin{aligned} (A_1 - \bar{\kappa}I)(A_1 - \kappa I) &= (A_1 - \bar{\kappa}I)Q_1 R_1 = \\ Q_1 Q_1^H (A_1 - \bar{\kappa}I)Q_1 R_1 &= Q_1 (A_2 - \bar{\kappa}I)R_1 = (Q_1 Q_2)(R_2 R_1). \end{aligned}$$

Így $Q = Q_1 Q_2$ éppen az $(A_1 - \bar{\kappa}I)(A_1 - \kappa I)$ mátrix QR -felbontásának Q tényezője, és ebből eredően valós.

Így ha a QR -algoritmus két lépését komplex konjugált eltolásokkal végezzük, az eredmény névlegesen valós lesz. A gyakorlatban azonban a kerekítési hiba miatt komplex mátrixot fogunk kapni, méghozzá olyat, amelyben a számok nagy képzetes összetevőt is tartalmazhatnak ([1]). Szerencsére létezik indirekt számítási mód Q és A_3 kiszámítására anélkül, hogy A_2 -t alkalmazni kellene.

Nézzük a következő algoritmust (implicit eltolás)

1. Számítsuk ki Q első q oszlopát
2. Legyen G olyan ortogonális mátrix, melyre $Ge_1 = q$
3. Legyen $B = G^T A G$
4. Ortogonális hasonlósági transzformációval hozzuk B -t Hessenberg-alakra, azaz találjunk olyan U -t, hogy $H = U^T B U$ Hessenberg-mátrix legyen

A G mátrixot most egy tükröző mátrixszal valósítsuk meg: $G := R(q - e_1)$, ugyanis erről már tudjuk, hogy $R(q - e_1)e_1 = q$. Az U mátrix megtalálásához használjuk fel az [1] könyvben leírt alábbi MATLAB algoritmusokat:

Hessenberg-alakra hozás Householder-tükrözésekkel:

```

U := I
for k = 1, 2, 3, ..., n - 1
    housegen(A[k + 1 : n, k], u, A[k + 1, k])
    v^T = u^T * A[k + 1 : n, k + 1 : n]
    A[k + 1 : n, k + 1 : n] = A[k + 1 : n, k + 1 : n] - u * v^T
    v = A[1 : n, k + 1 : n] * u
    A[1 : n, k + 1 : n] = A[1 : n, k + 1 : n] - v * u^T
    v = U[1 : n, k + 1 : n] * u
    U[1 : n, k + 1 : n] = U[1 : n, k + 1 : n] - v * u^T
end for k

```

Ha a Householder-transzformációt úgy vezetjük be, hogy $H = I - uu^T$, ahol $\|u\|_2 = \sqrt{2}$, akkor a *housegen* algoritmus adott x -hez megkeresi a Householder-mátrix u tükröző vektorát oly módon, hogy $\|u\|_2 = \sqrt{2}$, és ν -ben az előjelezett diagonális elem fog állni.

A Householder-transzformáció:

```

housegen(x, u, ν)
  ν = ||x||2
  if (ν = 0) u = √2e1; return; end if
  u = x/ν
  if (u1 ≥ 0) σ = 1
  else σ = -1
  end if
  u1 = u1 + σ
  u = u/√|u1|
  ν = -σν
end housegen

```

Mivel U ilyen alakú lesz az algoritmus elvégzése után:

$$\begin{pmatrix} 1 & 0 \\ 0 & U_* \end{pmatrix},$$

$GUe_1 = R(q - e_1)Ue_1 = R(q - e_1)e_1 = g$, vagyis GU olyan ortogonális mátrix, melynek első oszlopa g , valamint ez a mátrix hozza A_1 -et Hessenberg-alakúra. Azonban Q is egy olyan ortogonális mátrix, melynek első oszlopa g , és ő is Hessenberg-alakra hozza A_1 -et. Így felhasználva (4.14)-et $Q = GU$ és $H = A_3$. Tehát látjuk, hogy a fenti algoritmus közvetlen módon számítja ki a dupla eltolás eredményeit.

Egy működőképes algoritmushoz először azt kell megmutatnunk, hogyan tudjuk kiszámítani Q első oszlopát, valamint hogyan redukáljuk $G^T AG$ -t. Mindkét problémát leegyszerűsíti az a tény, hogy A_1 Hessenberg. Az egyszerűbb jelölés kedvéért most hagyjuk el az alsó indexet.

Mivel Q az $(A - \bar{\kappa}I)(A - \kappa I)$ QR -felbontásának ortogonális tényezője, Q első oszlopa arányos a szorzatmátrix első oszlopával (és ezáltal meghatározható az iránya):

$$\begin{pmatrix} a_{11} - \bar{\kappa} & a_{12} & * \\ a_{21} & a_{22} - \bar{\kappa} & * \\ 0 & a_{32} & * \\ 0 & 0 & * \end{pmatrix} \begin{pmatrix} a_{11} - \kappa & * \\ a_{21} & * \\ 0 & * \end{pmatrix} =$$

$$= \begin{pmatrix} a_{11}^2 - 2\Re(\kappa)a_{11} + |\kappa|^2 + a_{12}a_{21} \\ a_{21}a_{11} + a_{22}a_{21} - 2a_{21}\Re(\kappa) \\ a_{32}a_{21} \\ 0 \end{pmatrix} \equiv \begin{pmatrix} w \\ 0 \end{pmatrix}. \quad (4.19)$$

Láthatjuk, hogy Q első oszlopa csak három nemnulla elmet tartalmaz. Mivel G -t Householder-tükrözésként adtuk meg (4.4)-ben, így az csak A első három sorára és oszlopára fog vonatkozni. Így $G^T A G$ a következő formában fog megjelenni:

$$\begin{pmatrix} a & a & a & a & a & a & a & a \\ a & a & a & a & a & a & a & a \\ \hat{a} & a & a & a & a & a & a & a \\ \hat{a} & a & a & a & a & a & a & a \\ 0 & 0 & 0 & a & a & a & a & a \\ 0 & 0 & 0 & 0 & a & a & a & a \\ 0 & 0 & 0 & 0 & 0 & a & a & a \\ 0 & 0 & 0 & 0 & 0 & 0 & a & a \end{pmatrix}.$$

A következő lépés az, hogy ennek a mátrixnak az első oszlopában kiiktassuk a háztetős elemeket. Egymás után végrehajtott speciális 3×3 -as Householder-transzformációkkal a -2 -es átló mentén végül szép lassan számúzhetjük a háztetős elemeket. Az alábbi program ezt a redukálást alkalmazza:

A dupla eltolásos QR -algoritmus:

```

Számítsuk ki  $w$ -t (4.19)-ből
for  $k = 0$  to  $n - 2$ 
   $l = \min\{k + 3, n\}$ 
  if  $(k = 0)$  housegen( $w, u, t$ )
  else housegen( $A[k + 1 : l, k], u, A[k + 1, k], A[k + 2 : l, k] = 0$ )
  end if
   $v^T = u^T * A[k + 1 : l, k + 1 : n]$ 
   $A[k + 1 : l, k + 1 : n] = A[k + 1 : l, k + 1 : n] - u * v^T$ 
   $m = \min\{k + 4, n\}$ 
   $v = A[1 : m, k + 1 : l] * u$ 
   $A[1 : m, k + 1 : l] = A[1 : m, k + 1 : l] - v * u^T$ 
end for  $k$ 

```

l és m biztosítja, hogy a transzformációk A -n belül maradjanak

Eltolásra a legmegszokottabb megoldásként az alábbi mátrix sajátértékei kínálkoznak (ez Wilkinson nevéhez fűződik):

$$\begin{pmatrix} a_{n-1,n-1} & a_{n-1,n} \\ a_{n,n-1} & a_{nn} \end{pmatrix}.$$

Ki sem kell számítanunk őket, mert csak az alábbi mennyiségekre van szükségünk: ha $\lambda_1 = \lambda + i\mu$ és $\lambda_2 = \lambda - i\mu$, akkor

$$\lambda_1 + \lambda_2 = a_{n-1,n-1} + a_{n,n} = 2\lambda = 2\Re(\kappa)$$

és

$$\lambda_1 * \lambda_2 = a_{n-1,n-1}a_{n,n} - a_{n,n-1}a_{n-1,n} = \lambda^2 + \mu^2 = |\kappa|^2.$$

Ezek a formulák akkor is működnek, ha a sajátértékek valósak.

Szinguláris érték felbontás

Most példaként megmutatjuk, hogyan alkalmazható a QR -algoritmus az egyik legfontosabb mátrix-felbontás kiszámítására, a *szinguláris érték felbontás*-ra.

4.5. Tétel. *Legyen $A \in \mathbb{C}^{m \times n}$. Ekkor léteznek $U \in \mathbb{C}^{n \times n}$ és $V \in \mathbb{C}^{m \times m}$ unitér mátrixok, amelyekre fennáll, hogy*

$$V^H A U = \begin{pmatrix} \Sigma & 0 \\ 0 & 0 \end{pmatrix}, \quad \Sigma = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_r),$$

ahol $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r \geq 0$. A σ_i számokat az A mátrix szinguláris értékeinek nevezzük. Az U és a V oszlopait bal- illetve jobboldali szinguláris vektoroknak mondjuk.

Bizonyítás: [10] alapján.

$A^H A$ pozitív szemidefini mátrix, azaz minden sajátértéke nemnegatív, mivel ha $A^H A u = \lambda u$ úgy, hogy $\|u\|_2 = 1$, akkor $\lambda = u^H (A^H A) u = \|A u\|_2^2 \geq 0$. Legyenek a nemzérus sajátértékek $\sigma_1^2 \geq \sigma_2^2 \geq \dots \geq \sigma_r^2 \geq 0$. Ekkor létezik olyan U unitér mátrix, amelyre

$$U^H A^H A U = \begin{pmatrix} \Sigma^2 & 0 \\ 0 & 0 \end{pmatrix}.$$

Bontsuk fel a sajátvektorokból álló U mátrixot két részre: $U = (U_1 \ U_2)$, ahol U_1 a nemnulla, U_2 pedig a nulla sajátértékekhez tartozó sajátvektorokat tartalmazza. Ekkor $A U_2 = 0$, és

$$U_1^H A^H A U_1 = \Sigma^2 \quad \Rightarrow \quad \Sigma^{-1} U_1^H A^H \underbrace{A U_1}_{=0} \Sigma^{-1} = I.$$

Legyen $V_1 = AU_1\Sigma^{-1}$, így $V_1^H V_1 = I$ az előbbi alapján, azaz V_1 oszlopvektorai ortonormáltak. Egészítsük ki V_1 -et egy V_2 mátrixszal úgy, hogy $V = (V_1 \ V_2)$ unitér legyen: $V^H V = I$. Ekkor

$$V^H AU = \begin{pmatrix} V_1^H AU_1 & V_1^H(AU_2) \\ V_2^H AU_1 & V_2^H(AU_2) \end{pmatrix} = \begin{pmatrix} V_1^H V_1 \Sigma & 0 \\ (V_2^H V_1) \Sigma & 0 \end{pmatrix} = \begin{pmatrix} \Sigma & 0 \\ 0 & 0 \end{pmatrix},$$

ahol felhasználtuk, hogy $AU_2 = 0$ és $V_2^H V_1 = 0$. ♦

4.5. Többszörös sajátértékek esete

Azoknak az eseteknek a vizsgálatával, mikor nem teljesül a feltételünk, hogy a sajátértékek abszolút értékben különbözőek, vagy, hogy komplex konjugált gyökpárok vannak - tehát többszörös valós/komplex gyökök, vagy azonos abszolút értékű különböző gyökök fordulnak elő -, ebben a dolgozatban nem foglalkozunk. Azonban érdemes megjegyezni, hogy bizonyos módosításokkal ezekben az esetekben is alkalmazható a QR -algoritmus.

A téma iránt mélyebben érdeklődő Olvasóknak az alábbi könyvet és két cikket ajánljuk:

- G. W. Stewart: *Matrix Algorithms, Volume II: Eigensystems*
- Beresford N. Parlett: *Global Convergence of the Basic QR Algorithm on Hessenberg Matrices*, University of California, Berkeley, 1968
- A. Galántai, C. J. Hegedűs: *Hyman's method revisited*, Journal of Computational and Applied Mathematics
<http://dx.doi.org/10.1016/j.cam.2008.08.004>

Ez utóbbi cikkre hivatkozva igaz a következő

4.6. Tétel. *Redukálatlan felső Hessenberg-mátrixoknál többszörös sajátértékek esetén az eltolásos QR-algoritmus csak elsőrendű konvergencia-sebességgel rendelkezik, és az elérhető értékes jegyek száma d/m , ahol d decimális jegyre pontos az aritmetika és m a sajátérték multiplicitása.*

Köszönetnyilvánítás

Szeretnék köszönetet mondani témavezetőmnek, Hegedűs Csabának, hogy felkeltette érdeklődésemet a téma iránt és elvállalta segítségemet. Figyelemmel kísérte munkám minden mozzanatát, türelmesen magyarázott, rámutatott a hiányosságokra, kérdéseimmel bármikor fordulhattam hozzá.

Köszönettel tartozom Édesanyámnak, akitől sok biztatást, lelkesítést kaptam. Köszönöm Nagyszüleimnek az anyagi és lelki támogatást, melyekkel mindvégig segítettek tanulmányaimat. Köszönöm Testvéremnek, hogy amikor elcsüggedtem felvidított és lelket öntött belém.

Köszönöm gimnáziumi tanárainknak, hogy megszerettették velem a matematikát. Köszönetet mondok egyetemi oktatóimnak, akik segítettek, hogy betekintést nyerjek a matematika rejtelmeibe.

Köszönöm Barátomnak, hogy mindig biztatott, és ahol tudott, segítségemre volt.

Végül, de nem utolsó sorban köszönetet mondok évfolyamtársaimnak is, akikkel oly sokszor együtt készültünk a másnapi zh-kra, vizsgákra.

Irodalomjegyzék

- [1] G. W. Stewart: *AFTERNOTES goes to GRADUATE SCHOOL, Lectures on Advanced Numerical Analysis*, 1996
- [2] Hegedűs Csaba: *Sajátértékfeladatok, egyetemi jegyzet*, ELTE, IK
<http://numanal.inf.elte.hu/~hegedus/sefela.pdf>
- [3] Móricz Ferenc: *Numerikus módszerek az algebrában és analízisben*, Polygon Kiadó - SZTE Bolyai Intézet, 1997
- [4] Hegedűs Csaba: *Numerikus analízis, egyetemi jegyzet*, ELTE, IK
<http://www.inf.elte.hu/karunkrol/digitkonyv/Jegyzetek2008/numanal.pdf>
- [5] Rózsa Pál: *Lineáris algebra és alkalmazásai*, Műszaki Könyvkiadó, Budapest, 1976
- [6] David S. Watkins: *Fundamentals of Matrix Computations*, John Wiley & Sons, New York, 1991
- [7] Faragó István - Horváth Róbert: *Numerikus módszerek*, Budapest, 2011
<http://www.cs.elte.hu/~faragois/jegyzet.pdf>
- [8] Stoyan Gisbert, Takó Galina: *Numerikus módszerek I.*, TypoTex, Budapest, 2002
- [9] A. Galántai, C. J. Hegedűs: *Perturbations of invariant subspaces of unreduced Hessenberg matrices*, Computers and Mathematics with Applications, 2012
<http://dx.doi.org/10.1016/j.camwa.2012.04.001>
- [10] Hegedűs Csaba <http://numanal.inf.elte.hu/~hegedus/kiem.pdf>