# ZBATIME NUMERIKE PËR METODËN GMRES TË NËNHAPËSIRËS KRYLOV

**AMANDA ZEQIRI, LULJETA HODA, FATMIR HOXHA**

Universiteti i Tiranës, Fakulteti i Shkencave të Natyrës, Departamenti i Matematikës së Aplikuar

e-mail: amanda.zeqiri@fshn.edu.al

**Përmbledhje**

Në këtë punim sillet një pjesë e studimit të algjebrës lineare numerike, ku paraqiten tre probleme thelbësore të saj: sistemi linear i madh e i rrallë, vlerat vetjake dhe vlerat singulare. Fusha të ndryshme të studimit duhet të merren me matrica të mëdha dhe të rralla të të dhënave, madhësia e të cilave shfaq vështirësi për një sërë metodash. Pikërisht metodat e nënhapësirës Krylov mund të përdoren për zgjidhjen e këtyre lloj problemesh. Një hyrje e shkurtër paraqitet në fillimisht për metodën Krylov dhe përshkrimin singular, përkatësisht vlera minimale dhe maksimale. Më konkretisht shqyrtohet Metoda e Përgjithshme e Mbetjes Minimum ($GMRES$), e cila është pjesë e metodave të nënhapësirës Krylov. Kjo metodë përdoret për sistemet lineare jo Hermitiane në rastin e një matrice të madhe e të rrallë. Gjithashtu paraqitet mënyra se sa shpejt $GMRES$ konvergjon dhe si iteracioni Arnoldi lokalizon vlerat vetjake. Shpesh konvergjenca e metodës mund të jetë shumë e ngadaltë dhe procesi bëhet i pazbatueshëm, kështu që hyn në punë nevoja për parakushtëzim. Qasja është që së pari këto metoda të zhvillohen nga ana e një algoritmi dhe më pas të tregohet se si përdoret në zbatime praktike me anë të software-it të llogaritjeve teknike dhe numerike, Matlab.

**Fjalëkyçe:** Metodat Krylov, iteracioni Arnoldi, metoda $GMRES$, parakushtëzimi, vlerat singulare.

**Abstract**

This paper presents a part of numerical linear algebra study, which presents three essential problems: the large and sparse linear system, eigenvalues and singular values. Different fields of study have to deal with large and sparse data matrices, the size of which presents difficulties for a variety of methods. It is the Krylov subspace methods that can be used to solve these types of problems. A brief introduction is presented at the beginning for the Krylov method and the singular description, respectively the minimum and maximum values. More specifically, the Generalized Minimal Residual Method (GMRES) is considered, which is part of the Krylov subspace methods. This method is used for non-Hermitian linear systems in the case of a large and sparse matrix. It also shows how quickly GMRES converges and how Arnoldi iteration localizes eigenvalues. Often the convergence of the method can be very slow and the process becomes inapplicable, so the need for preconditioning comes into play. The approach is to first develop these methods by an algorithm and then show how they are used in practical applications by using technical and numerical calculations software, Matlab.

**Keywords:** Krylov methods, Arnold iteration, GMRES method, preconditioning, singular values.

## Hyrje

Në lidhje me "zhvillimin dhe praktikimin e shkencës dhe inxhinierisë në shekullin e 20-të", metodat e nënhapësirës Krylov konsiderohen si një nga 10-të klasat më të rëndësishme të metodave numerike për zgjidhjen e sistemeve lineare algjebrike (Corso, Menchi & Romani, 2015). Në vitin 1931 një punim nga shkencëtari rus Aleksei Nikolaevich Krylov, paraqiti mënyrën si të përdorte sekuencat e formës $\{r, A\,r, A^2r, \ldots\}$ për të ndërtuar polinomin karakteristik të një matrice (Krylov, 1931). Këto metoda i studiojmë veçanërisht në sistemet lineare të mëdha dhe të rralla, në aplikimet që hasin në fushat konkrete në shkencë dhe inxhinieri.

Metodat Krylov japin një klasë të madhe të algoritmeve iterative që punojnë me matrica dhe vektorë. Ato kanë shumë aplikime, ku përmendim problemet me katrorët më të vegjël linearë, zgjidhjen e sistemeve lineare, problemet e vlerës vetjake dhe problemet e vlerës singulare. Nënhapësirat Krylov përbëhen nga mbetja fillestare dhe nga vektorët e formuar nga shumëzimi iterativ i mbetjes fillestare me matricën e sistemit.

Dy grupimet kryesore të metodave iterative janë ato stacionare dhe më të përgjithshme të nënhapësirës Krylov. Këto metoda bazohen nga ana matematikore në metodat e projeksionit. Ideja kryesore e nënhapësirës Krylov dhe në përgjithësi e një procesi projeksioni, është gjetja e një zgjidhjeje të përafërt të një sistemi potencialisht shumë të madh $Ax = b$ duke zgjidhur një sistem me dimension shumë më të vogël. Algoritmet e bazuara në nënhapësirën Krylov llogaritin në mënyrë iterative një përafrim të $x$. Zakonisht, marrim një vlerësim të përafërt të $Ax = b$ me $x_0$, atëherë le të jetë $x = x_0 + x_{\mathrm{m}}$. Bashkësia e vektorëve:

$$K_{\mathrm{m}} = \{r_0, A\,r_0, A^2\,r_0, \ldots, A^{\mathrm{m}}\,r_0\} \qquad (1.1)$$

ku $m < rank(A)$, quhet matrica Krylov e madhësisë $m$. Vlerësimi i $x_{\mathrm{m}}$ formon një përafrim të nënhapësirës Krylov $K_{\mathrm{m}}$:

$$K_{\mathrm{m}}\,(A, r_0) = span\,\{r_0, A\,r_0, A^2\,r_0, \ldots, A^{\mathrm{m}}\,r_0\} \qquad (1.2)$$

Për të zgjidhur një sistem linear, metoda të njohura si eliminimi i Gausit ose iteracione të tjera të thjeshta siç është iteracioni i Jakobit mund të vendosen të parët në konsideratë, por përballë matricave të përmasave të mëdha, këto metoda nuk janë aq të dobishme. Gjithashtu, këto matrica mund të jenë të papërcaktuara dhe asimetrike. Në rastin e një sistemit të madh dhe linear, përdorimi i metodave

Krylov është me të vërtetë efektiv (Shitao Fan, 2018). Kemi që kur matrica është singulare, metodat Krylov mund të dështojnë. Këta vektorë nuk janë ortogonalë, por mund të merren për shembull me mënyrën Gram-Schmidt të ortogonalizimit. Metoda e hapësirës Krylov për zgjidhjen e një sistemi linear $Ax = b$ është një metodë iterative duke filluar nga përafrimi fillestar dhe mbetja përkatëse.

**Dekompozimi në vlera singulare**

Në algjebrën lineare, një rol të veçantë luajnë vektorët e ndryshëm nga zero, që gjatë një transformimi kalojnë në vektorë që ndryshojnë nga skalarët. Këta quhen vektorë vetjakë të transformimit. Marrim një matricë katrore $A$ ku $v \neq 0$ i tillë që:

$$Av = v\lambda \qquad (2.1)$$

$\lambda$ është një vlerë vetjake e matricës $A$:

$$det(A - \lambda I) \equiv p(\lambda) = 0 \qquad (2.2)$$

kemi që $v$ është një vektor vetjak përkatës. Shumë probleme inxhinierie kërkojnë vetëm vlerën vetjake më të madhe apo vlerën vetjake më të vogël të një matrice katrore. Konkretisht për një urë ose kolonë mbështetëse, vlera vetjake më e vogël zbulon ngarkesën maksimale dhe vektori vetjak përfaqëson formën e objektit në pjesën e dështimit nën këtë ngarkesë.

Rrënja katrore e vlerave vetjake të $A^*A$ quhen vlerat singulare dhe i kemi shënuar me $\sigma_i \geq 0,\ i = 1,2,\ldots,n$. Këto $\sigma$ mund të përcaktohen si $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n \geq 0$. Vektorët vetjak $v_i$ përkatës të vlerës singulare për $i = 1,2,\ldots,n$, quhen vektorë singularë të djathtë të $A$. Një vlerë singulare dhe një çift vektorësh singularë të një matrice $A$ janë skalar jonegativ $\sigma$ dhe dy vektorët jozero $u$ dhe $v$ janë:

$$Av = \sigma u \text{ dhe } A^*u = \sigma v \qquad (2.3)$$

Në mënyrë të përmbledhur paraqesim zbërthimin në vlera singulare:

- Vlera singulare e $A \leftrightarrow$ vlera vetjake e $A^TA$ ose $AA^T$
- Vektori singulare i djathtë i $A \leftrightarrow$ vektori vetjak i $A^TA$
- Vektori singulare i majtë i $A \leftrightarrow$ vektori vetjak i $AA^T$
- Numri i vlerave singulare jozero të $A$ është i barabartë me rangun e matricës $A$.

Si konkluzion, nëse arrihet të zgjidhet problemi i vlerave vetjake për një matricë $A$, atëherë automatikisht dhe problemi i vlerave singulare është i zgjidhur.

$DVS$-ja konsiderohet mjeti më i besueshëm dhe i përdorur gjerësisht në zgjidhjen e problemeve algjebrike lineare numerike. Vlerat singulare dhe zbërthimet e vlerës singulare konsiderohen shumë të rëndësishme në analizimin e të dhënave. Problemi i vlerave vetjake të përgjithësuara, katrorët më të vegjël të përgjithësuar përfshirë dhe fushën e njehsimeve matricore imponojnë zgjerimin e përkufizimit të vlerave singulare. Ato luajnë një rol thelbësor kur bëhet fjalë për të siguruar ekzistencën dhe unitetin e zgjidhjeve Krylov, pra nuk janë thjesht një objekt i analizave të konvergjencës (Ipsen & Meyer, 1997).

**Parakushtëzimi**

Në matematikë kemi të bëjmë me aplikimin e një transformimi parakushtëzues, që kushtëzon një problem të dhënë në një formë që është më e përshtatshme për metodat e zgjidhjes numerike. Problemi i parakushtëzuesve zakonisht zgjidhet me një metodë iterative, që zbatohen për zgjidhjen e sistemeve lineare me përmasa të mëdha (Yousef Saad, 2000).

Ekzistojnë tre lloje të përgjithshme të parakushtëzimit:

- Parakushtëzimi i majtë nga një matricë $M_1 \approx A$:

$M_1^{-1} A x = M_1^{-1} b$

- Parakushtëzimi i djathtë nga një matricë $M_2 \approx A$:

$AM_2^{-1} y = b, x = M_2^{-1} y$

Këtu përfshihet një zëvendësim y për variablin origjinal $x$.

- Parakushtëzimi i ndarjes me një çift matricash $M_1$, $M_2$ të tilla që $M_1 M_2 \approx A$:

$M_1^{-1} A M_2^{-1} y = M_1^{-1} b, x = M_2^{-1} y$

Këtu përfshin metodat e majta dhe ato të djathta duke vendosur përkatësisht $M_1 u = I$ ose $M_2 = I$.

$M$ është një matricë përafruese për $A$ −në, i zgjedhur në një mënyrë që zëvendësimi i $M^{-1}A$ ose $AM^{-1}$ në vend të $A$ të çon në një sjellje më të mirë të konvergjencës.

**Metodat për llogaritjen e $DVS$**

Proceset Lanczos dhe Arnoldi paraqesin fillesat e metodave të nënhapësirës Krylov për të zgjidhur sistemet lineare të përcaktuara pozitivisht, rreth vitit 1950. Algoritmi Krylov i Lanczos përdoret për llogaritjen e tre çifteve singulare të vlerave singulare dhe vektorëve singularë. Iteracioni Arnoldi është metoda më e efektshme për llogaritjen e disa çifteve vejake për matricat e mëdha e të rralla. Këto dy metoda janë shumë të ndjeshëm ndaj turbullimeve në iteracione dhe kërkojnë zgjidhje shumë të sakta të sistemeve lineare.

Ndërtimi i bazave ortogonale studiohet me detaj në hapësirat Krylov. Procedura themelore paraqitet nga një variacion i algoritmit të njohur Gram-Schmidt të një matrice $A \in R^{\mathrm{nxn}}$ në formën e një matrice Hesenberg. Ky lloj ndërtimi është baza për metodat e përgjithshme të nënhapësirës Krylov.

Disa metoda të njohura të nënhapësirë Krylov janë (C. T. Kelley, 1995):

(1) algoritmi i Hermitit Lanczos;

(2) metoda e Arnoldit dhe variacionet e saj;

(3) algoritmi i jo Hermitit Lanczos.

Më i përdoruri në hapësirën Krylov është procesi i Arnoldit, i cili u paraqit si një mjet për të zvogëluar një matricë të dendur me një transformim unitar në formën e matricës Hesenberg. Kjo metodë është pjesë e projeksionit ortogonale mbi $K_{\mathrm{m}}$ për matricat e përgjithshme jo Hermitiane dhe zgjedh $h_{\mathrm{ij}}$ në mënyrë që baza të jetë ortonormale (Auzinger & Melenk, 2017). Vlerat vetjake të matricës Hesenberg mund të sigurojnë përafrime të sakta në disa vlera vetjake të matricës origjinale.

Kjo lloj strategjie të çon në një teknikë për përafrimin e vlerave vetjake të matricave të mëdha e të rralla, e cila u përdor më pas në zgjidhjen e sistemeve lineare të mëdha e të rralla të ekuacioneve. Një matricë $A \in R^{\mathrm{nxn}}$ dhe një vektor i caktuar $r_0 \in R^{\mathrm{n}}$ përcakton një sekuencë të nënhapësirave të Krylovit $K_{\mathrm{m}} = K_{\mathrm{m}}(A, r_0) = span\{r_0, A\,r_0, A^2\,r_0, \dots\dots, A^{\mathrm{m}}\,r_0\},\ m = 0, 1, 2, \dots$

Më poshtë është paraqitur algoritmi Arnoldi për vlerat vetjake (Auzinger & Melenk, 2017).

**Algoritmi 1**. Procedura Arnoldi

1. Zgjidhet një vektor fillestar për nënhapësirën Krylov dhe normalizohet $v_1$.

2. Aplikohen $m$ hapat të iteracionit Arnoldi.

për $j = 1, 2, \dots, m,$

$w_{\mathrm{j}} = Av_{\mathrm{j}}, i = 1,2, \dots, j$

$h_{\mathrm{ij}} = (w_{\mathrm{j}}, v_{\mathrm{i}})$

$w_{\mathrm{j}} = w_{\mathrm{j}} - h_{\mathrm{ij}} v_{\mathrm{i}}$

$$h_{j+1,j} = \|w_j\|_2 \text{ dhe}$$

$$v_{j+1} = w_j / h_{j+1,j} .$$

3. Gjenden çiftet vetjake $(\lambda_j, u_j)$ të $H_j$ dhe zgjedhim $r$ çiftet Ritz $(\lambda_j, x_j)$ ku $x_j = V_j u_j$, si përafrimet e çiftit vetjak të dëshiruar.

4. Normat e mbetjes mund të kontrollohen për konvergjencë. Nëse është e nevojshme, zgjidhet një vektor i ri fillestar për $v_1$ dhe kalon në 2.

---

Të ndërtuarit e një baze ortonormale $\{v_1, \ldots, v_m\}$ të $K_m$ është një teknikë themelore për atë që vijon dhe për algoritme të përgjithshme të bazuara në sekuencat Krylov. Në parim ndërtojmë $\{v_1, \ldots, v_m\}$. Aplikojmë algoritmin Gram-Schmidt ose një tjetër procedurë ortonormalizimi ekuivalente tek vektorët Krylov $r_0, Ar_0, \ldots$ (kolonat e $K_m$). Vektorët ortonormal që rezultojnë $v_j$ quhen vektorët Arnoldi. $GMRES$ është një mënyrë për të implementuar këtë metodë.

Algoritmi Lanczos përfaqëson një algoritëm nga Cornelius Lanczos, i cili shihet si adaptim i metodave të fuqisë për të gjetur $m$ me prirje drejt vlerave vetjake ekstreme dhe vektorëve vetjak të një matrice Hermitiane $nxn$. Konkretisht e kemi të paraqitur si një thjeshtëzim të metodës së Arnoldit për rastin e veçantë kur matrica është simetrike, ku matrica Hesenberg $H_m$ shndërrohet në tridiagonale simetrike. Ky algoritëm ndërton një bazë ortonormale për nënhapësirën Krylov në matricat hermitiane. Për të gjetur vlerat vetjake të një matricë simetrike $A$ bashkohet algoritmi Lanczos për ndërtimin e nënhapësirës Krylov me procedurën Rayleigh-Ritz.

Algoritmi i Lanczos për të gjetur vlerat vetjake dhe vektorët vetjakë kur kemi $A = A^T$.

---

**Algoritmi 2.** Algoritmi Lanczos

---

$v_1 = r_0 / \|r_0\|_2,\ \beta_1 = 0, v_0 = 0$

për $j = 1,2, \ldots, m$

$w_j = Av_j - \beta_j v_{j-1}$

$\alpha_j = h_{jj} = (w_j, v_j)$

$w_j = w_j - \alpha_j v_j$

$\beta_{j+1} = h_{j,j-1} = \|w_j\|_2$

nëse $\beta_{j+1} = 0$ atëherë algoritmi ndalon

$v_{j+1} = w_j/\beta_{j+1}$

Njehsoj çiftin vetjak $(\lambda_j, u_j)$ të $H_j$ dhe zgjedhim $r$ çiftet Ritz (λ□, x□), ku $x_j = V_j u_j$, si përafrimet e çiftit vetjak të dëshiruar.

Prandaj dhe Algoritmi Lanczos e transformon problemin e vlerave vetjake të $A$ në problemin e vlerave vetjake të $H_k$, i cili zgjidhet duke aplikuar metodën $QR$. Në këtë mënyrë, vlera vetjake më e vogël e $H_k$ do të korrespondojë me vlerën më të vogël të matricës $A$.

**Metoda $GMRES$ (Metoda e Përgjithësuar e Mbetjeve Minimale)**

Metoda e mbetjes minimale së përgjithsuar u zhvillua nga Yousef Saad dhe Martin H. Schultz (1986), i cili jepet si një përgjithësim i metodës $MINRES$ bazuar në procesin Arnoldi për llogaritjen e vlerave vetjake të një matrice. Në matematikë, $GMRES$ është një metodë iterative për zgjidhjen numerike të një sistemi josimetrik të ekuacioneve lineare. Modifikimi i procedurës Gram-Schmidt paraqet një ndër format më të njohura të $GMRES$ (Saad & Schultz, 1986).

Kjo metodë është e njohur për zgjidhjen e një sistem të rrallë të ekuacioneve lineare $Ax = b$, veçanërisht, kur një zgjidhje e përafërt është e përshtatshme (Saad & Schultz, 1986). GMRES konsiderohet të jetë e qëndrueshme për zgjidhjen e sistemeve lineare josimetrike, sidoqoftë veprimet për iteracion rriten me numrin e iteracioneve. Gjithashtu kjo metodë përfshin një nënmatricë të sipërme të matricës Hesenberg të gjeneruar nga procesi Arnoldi, e cila lejon marrjen e kufijve që tregon se norma e mbetjes zvogëlohet në $0$ kur vlera më e vogël singulare e këtij matricë trekëndore shkon në $0$.

Zbatimi standard i tij bazohet në procesin Arnoldi. Metoda përafron zgjidhjen me anë të vektorit në një nënhapësirë Krylov me mbetje minimale. Iteracioni Arnoldi përdoret për të gjetur këtë vektor. Përdorim një zbërthim $QR$ në vend të rrotullimit të Givens. Pra do përqendrohemi në dy algoritme për matricën josimetrike $GMRES$. E para e cila zgjidh sistemet e ekuacioneve $Ax = b$ dhe Arnoldi, i cili llogarit vlerat vetjake të $A$-së. Më poshtë është paraqitur algoritmi $GMRES$:

**Algoritmi 3.** $GMRES$

1. Llogaritet $r_0 = b - Ax_0$, $\beta := \|r_0\|_2$ dhe $v_1 := r_0 / \beta$

merret një vlerësim i përafërt i ekuacionit dhe inicializohet vektori primar i nënhapësirës Krylov.

2. Për $j = 1,2,..,m$ bëj:

3. Llogaritet $w_j = Av_j$

ndërtohet vektori tjetër për përfshirje në nënhapësirën Krylov.

4. Për $i = 1,...,j$ bëj:

5. $h_{ij} = (w_j, v_i)$

6. $w_j = h_{ij}v_j$

Schmidt ortogonalizon vektorin.

7. Fund

8. $h_{j+1,j} = ||w_j||_2$, *nëse* $h_{j+1,j} = 0$, vendos $m = j$ dhe ndërpritet

provohet nëse nënhapësira është nënhapësirë invariante.

9. $v_{j+1} = w_j / h_{j+1,j}$ normalizohet vektori

10. Fund

11. Përcaktohet matrica Hesenberg

$(m+1) * m, H_m = \{h_{ij}\}, 1 \leq i \leq m+1, 1 \leq j \leq m$

12. Llogaritet $y_m$ minimizuesi i $\| \beta e_1 - H_m y_m \|_2$ dhe $x_m = x_0 + V_m y_m$. Përdoret katrori linear më i vogël për të minimizuar mbetjen.

Përkufizohet si $y_m$ zgjidhja e problemit të minimizimit

$min \| \beta e_1 - H_m y \|_2$.

Në këtë mënyrë, $\mathrm{x}$ optimal jepet nga $\mathrm{x_m = x_0 + V_m y_m}$. Vëmë re që ky problem minimizimi është një problem i katrorëve të vegjël i madhësisë $\mathrm{m+1}$ dhe matrica e koeficienteve të tij është Hesenberg e sipërme. Problemi i minimizimit zgjidhet duke përdorur faktorizimin QR të matricës $H_m$.

**Llogaritja e vlerave vetjake ekstremale**

Kur kemi të bëjmë me matricat e mëdha shohim vështirësi dhe kompleksitet të lartë gjatë llogaritjeve të ndryshme. Kështu që mundësia më e mirë është aplikimi i algoritmeve nëpërmjet gjuhëve të ndryshme të programimit. Në këtë pjesë do trajtojmë problemin e vlerave vetjake për një matricë të madhe të rrallë të përmasave $479 \times 479$. Nëpërmjet Matlab do shpjegojmë se si gjenerohen vlerat vetjake të kësaj matrice gjë e cila manualisht do ishte e pamundur. Më poshtë kemi përdorur metodën e Arnoldit për të bërë më pas edhe lidhjen me metodat $GMRES$.

Kodi në Matlab merr si input matricën dhe gjeneron si output vlerat vetjake dhe paraqitjen grafike të tyre. Duke qenë se numri i vlerave vetjake është i madh për shkak të dimesioneve të matricës kemi zgjedhur që të shfaqim respektivisht $6$ vlerat më të mëdha dhe $6$ më të voglat, ku dy prej vlerave paraqiten të njëjta. Vlera më e vogël vetjake është vlera $0.0114$, shumë afër zeros. Gjë që shihet dhe grafikisht, ku vlera përkatëse është pozicionuar shumë afër origjinës së koordinatave të boshtit koordinativ. Shfaqim grafikisht të gjitha vlerat vetjake të shënuara me $d$, i mbivendosim në grafik $6$ vlerat më të mëdha të shënuara me ngjyrë të kuqe dhe $6$ vlerat më të vogla të shënuara me ngjyrë jeshile. Paraqitja grafike jepet në figurën e mëposhtme:

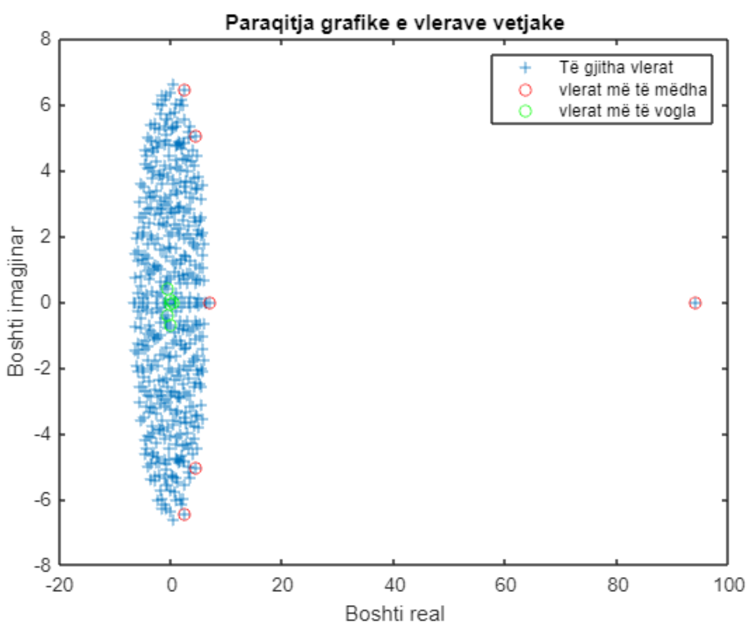


**Figura $6.1$**. Vlerat vetjake ekstremale

Nga figura $6.1$ shohim që vlerat vetjake janë të shpërndara pothuajse “uniformisht” në formën e një elipsi, e cila vizualisht nuk duket aq qartë për shkak se ndarja e boshtit real dhe imagjinar nuk është e njëjtë. Vlerat vetjake më të vogla janë të pozicionuar shumë afër origjinës së koordinatave, zonë në të cilën edhe distanca është shumë afër $0$. Ndërsa vlerat vetjake më të mëdha janë të pozicionuara në skaje, zona në të cilat distanca është me “magnitudën” maksimale.

Vlerat vetjake në përgjithësi janë në trajtë komplekse $x + iy$ dhe afishimi i tyre në graf bëhet duke konsideruar $OX$ boshtin real dhe $OY$ boshtin imagjinar. Një diskutim që bëhet në lidhje me përcaktimin e vlerave më të vogla dhe më të mëdha vetjake është fakti se cili parametër duhet të konsiderohet për të përcaktuar “madhësinë” e vlerës.

Kemi $3$ kritere që përdoren:

- konsiderohet vetëm pjesa reale. Renditja e vlerave vetjake bëhet vetëm në bazë të $x$.
- konsiderohet vetëm pjesa imagjinare. Renditja e vlerave vetjake bëhet vetëm në bazë të $y$.
- konsiderohet magnituda. Pra renditja përcaktohet nga madhësia : $\sqrt{x^2 + y^2}$.

Në përzgjedhjen tonë të $6$ vlerave më të mëdha dhe më të vogla kemi përdorur si kriter pikërisht madhësinë e fundit.

**Zgjidhja iterative e sistemit linear**

Fillimisht, do të shfaqim metodën $GMRES$ për zgjidhjen e sistemeve të mëdha. Për të zgjidhur $Ax = b$, $gmres()$ llogarit zgjidhjen e përafërt $x_{\mathrm{m}}$ nga nënhapësira Krylov që minimizon normën e mbetjes, e cila zakonisht optimizohet në normën euklidiane. Algoritmi mund të zbatohet në çdo normë të induktuar nga një produkt i brendshëm. Ky veprim nuk ndikon as spektrin e $A$-së dhe as nënhapësirën $K_{\mathrm{m}}(A, b)$, por mund të ndryshojë ndjeshëm largimin e $A$ nga normaliteti dhe kushtëzimin e vlerave vetjake.

$GMRES$ shpesh ka sjellje superlineare të konvergjencës, pra shkalla e konvergjencës duket se përmirësohet ndërsa procedura vazhdon (Kelley, 1995). Në përgjithësi, sjellja e konvergjencës do të varet nga struktura vetjake e matricës dhe nga pseudo vlerat vetjake. Kjo metodë konvergjon sa më shpejt që të kemi marrë një numër iteracionesh të barabarta me numrin e vlerave vetjake. Konkretisht, mund të shikojmë efektin e strukturës vetjake në zvogëlimin e mbetjes në secilin iteracion. Studiojmë se si evoluon mbetja, duke mbajtur parasysh formën polinomiale të mbetjeve të $GMRES$.

Konkretisht krijohet një matricë e rastësishme dhe e rrallë $A$ me përmasa $479x479$, me dendësi $50\%$ dhe elementë jo-zero në diagonalen kryesore. Dhe krijojmë një vektor të rastit $b$ me përmasa $479x1$ për anën e djathtë të ekuacionit matricor $Ax = b$. Në tabelën e mëposhtme paraqesim nëse jemi në rastin e një algoritmi që konvergjon ose jo:

| Toleranca | Maxit | Restart | Mbetja relative | Rezultati |
|---|---|---|---|---|
| $1e-6$ | 10 | $-$ | $0.55$ | Nuk konvergjon |
| $1e-4$ | $100$ | $-$ | $0.28$ | Nuk konvergjon |
| $1e-4$ | $300$ | $100$ | $7.2e-05$ | Konvergjon |

**Tabela $7.1$:** $gmres()$ për zgjidhjen e sistemeve të mëdha pa parakushtëzim

Sistemi linear katror zgjidhet duke përdorur $gmres()$ me kushtet e paracaktuara dhe më pas rregullohet toleranca, numri i iteracioneve të përdorura gjatë procesit të zgjidhjes. Rezultati i shfaqur përfshin edhe vlerën e gabimit relativ të mbetjes. Sidoqoftë, $gmres()$ ka gjithashtu një input që kontrollon numrin e iteracioneve të brendshme. Zakonisht përdoren $10$ iteracione dhe një tolerancë prej $10^{-6}$. Në rastin e matricës sonë shohim që algoritmi nuk konvergjon në ato 10 iteracione. Edhe me një tolerancë më të lirshme dhe me shumë iteracione, gabimi i mbetjeve nuk përmirësohet shumë. Edhe pse implementimi në Matlab na paraqet një mbetje ende të madhe, është një tregues që nevojiten më shumë iteracione ose një matricë parakushtëzuese.

Fakti nëse konvergjon ose jo metoda përfaqëson shkallën me të cilën një metodë iterative i afrohet zgjidhjes së një sistemi linear dhe ofron një lidhje për normat e mbetjes të $gmres()$. Kështu sistemi do zgjidhet përsëri duke përdorur një vlerë rifillimi (restart) $100$ dhe një vlerë maksimale iteracionesh prej $300$. Në vend që të bëhen $100$ iteracione një herësh, $gmres()$ kryen $100$ iteracione midis rifillimit dhe përsërit këtë veprim 300 herë. Ndryshe nga rastet e tjera ku numri i iteracioneve është shumë larg numrit real të vlerave vetjake të matricës $A$. Konvergjenca e shpejtë në rastin e tretë ndodh, sepse vlerat vetjake më të vogla të $A$ janë grumbulluar larg origjinës dhe $A$ nuk është shumë larg normalitetit. Gjithashtu specifikimi i një vlere të madhe të ristartimit për $gmres()$ i mundëson metodës të konvergjojë në një zgjidhje brenda numrit të lejuar të iteracioneve.

**Parakushtëzimi $gmres()$ pa Restart**

Për të zgjidhur sistemin linear shqyrtojmë efektin e parakushtëzimit të matricës pa restart. Konkretisht marrim një matricë të rrallë josimetrike $(479x479)$ dhe përcaktohet $b$ në mënyrë që zgjidhja e vërtetë e $Ax = b$ është një vektor me të gjitha hyrjet $1$. Përcaktohet toleranca $1e-11$ dhe numri maksimal i iteracioneve, që është $11$-të. Kemi që konvergjenca e metodave të nënhapësirës Krylov për ekuacionet lineare varet në një shkallë të madhe nga shpërndarja e vlerave vetjake.

Metoda GMRES pa Restart ka të bëjë dhe me faktin që një vektor i përafruar i zgjidhjeve do të ruhet në çdo iteracion derisa metoda të konvergjojë dhe gabimi relativ do të bëhet më i vogël sesa limiti i lejuar. Problemi i metodës pa Restart qëndron në rritjen e kompleksitetit të llogaritjeve. Siç e theksuam më sipër një mënyrë për të mos u zgjatur procesi vendoset një numër maksimal iteracionesh por kjo nuk garanton që metoda konvergjon, ndalon procesimin përtej një numri iteracionesh brenda të cilit duhet të ishte arritur konvergjenca.

Pra përdorim $gmres()$ për të gjetur zgjidhjen me tolerancën dhe numrin e iteracioneve të kërkuar. Gjithashtu specifikojmë $5$ outputet për të përcjellë informacionin për procesin e zgjidhjes:

- $fl0$ paraqet një 'flag' dhe tregon nëse algoritmi konvergjon apo jo.
- $x0$ paraqet zgjidhjen e llogaritur e $A * x_0 = b$.
- $it0$ është një vektor me dy elementë inner dhe outer, i cili tregon numrin e brendshëm dhe të jashtëm të iteracioneve kur $x_0$ është llogaritur.
- $rr0$ paraqet mbetjen e zgjidhjes së llogaritur $x_0$.
- $rv0$ është një vektor që përfaqëson historinë e mbetjeve për $||Ax - b||$.

Në rastin e problemit të dhënë më sipër kemi:

| fl0 | fl1 | rr0 | rr1 | it0 | | it1 | |
|---|---|---|---|---|---|---|---|
| 1 | 0 | 0.6513 | 1.3557e-13 | 1 | 25 | 1 | 6 |

**Tabelë $8.1$:** Outputet në procesin e zgjidhjes

$fl0$ merr vlerën $1$, sepse $gmres()$ nuk konvergjon në tolerancën e kërkuar $1e-11$ brenda $25$ iteracioneve të kërkuara. Zgjidhja e përafërt më e mirë që GMRES kthen është paraqitur nga $it0\ (2) = 25$. Kështu që për të ndihmuar në konvergjencë, mund të specifikojmë një matricë parakushtëzimi.

Duke qenë se $A$ është matricë josimetrike, përdorim $ILU$ për të gjeneruar parakushtëzimin $M = LU$ dhe përcaktojmë një tolerancë 'drop' për të mos lejuar hyrjet jodiagonale me vlerë më të vogël se $1e-6$. Më pas zgjidhim sistemin e parakushtëzuar $M^{-1}Ax = M^{-1}b$ duke i përcaktuar $L$ dhe $U$ si inpute të GMRES. Kur specifikohet një parakushtëzim, $gmres()$ llogarit normën e mbetjes të sistemit të parakushtëzuar për outputet $rr1$ dhe $rv1$.

Përdorimi i parakushtëzimit $ILU$ prodhon një mbetje relative më të vogël se toleranca e kërkuar $1e-11$ në iteracionin e gjashtë të jashtëm dhe ecurinë e $gmres()$ mund ta shohim duke vënë në grafik mbetjet relative për çdo iteracion. Konkretisht mbetja e parë $(rv1(1))$ është $norm(U \setminus (L \setminus b))$, ku $M = L * U$ dhe mbetja e fundit është $norm(U \setminus (L \setminus (b - A * x1)))$.

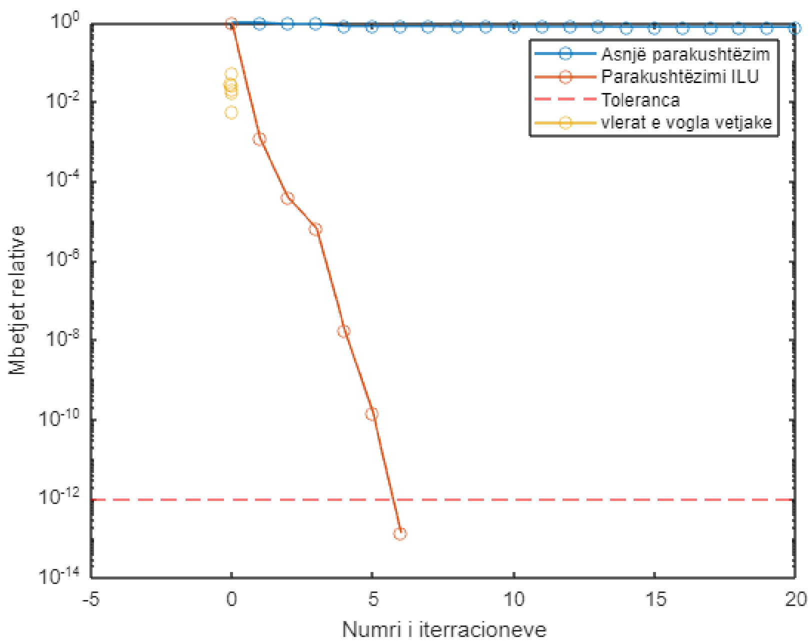


**Figura 8. 1** Parakushtëzimi $gmres()$ pa Restart

Në figurën e mësipërme paraqitet ulja e vlerave të mbetjes me rritjen e iteracioneve, ku kemi dhe kalimin e kufirit të vendosur. Rasti i pakushtëzuar paraqet vetëm vlerat e metodës, e cila nuk sjell një konvergjencë të sistemit linear. Grafiku me ngjyrë portokalli tregon sesi ndryshon hap pas hapi gabimi relativ i metodës. Kufiri i poshtëm paraqet kufirin në të cilin metoda konvergjon.

Gjithashtu kemi vlerat më të vogla vetjake, të cilat nuk janë shumë afër $0$-s prandaj metoda konvergjon ngadalë nëse vlerat më të vogla vetjake janë afër $0$-s dhe konvergjojnë më shpejt nëse këto vlera nuk janë shumë afër $0$-s. Në rastin e problemit tonë metoda konvergjon shpejt me 6 iteracione dhe gabimi relativ është më i vogël se $10^{-11}$. Nga grafiku shohim që gabimi i metodës bëhet gjithnjë e më i vogël dhe në iteracionin e $6$-të metoda konvergjon shpejt në sajë të parakushtëzimit ILU.

## Përfundime

Arritëm në përfundimin që përcaktimi i një vlere të madhe të rifillimit për GMRES i bën të mundur metodës të konvergjojë në një zgjidhje brenda numrit të lejuar të iteracioneve. Gjithashtu treguam se vlerat e vetjake mund të jenë shumë të dobishme për zgjidhjen e sistemeve dhe nga shembujt e trajtuar më sipër pamë që shkalla e konvergjencës udhëzohet nga vlerat vetjake më të vogla, të cilat sjellin ngadalësimin e shpejtësisë së saj.

Qëllimi ynë është të kuptojmë se kur dhe pse metoda konvergon më shpejt sesa në një fazë e mëparshme e procesit të iteracionit. Një nga teknikat që sjell në një konvergjencë më të shpejtë është parakushtëzimi i matricës. Konkretisht kemi përmendur si parakushtëzim ILU-në, i cili përmirëson vetëm numrin e kushtit të sistemit matricor duke e sjellë më afër matricës njësi. Është e qartë se zgjedhja e një parakushtëzuesi të gabuar mund të ndalojë konvergjencën dhe kështu të gabojë përafrimin e zgjidhjes. Kjo metodë përdoret zakonisht kur matrica është e keqkushtëzuar, prandaj dhe implementimi ndikon në numrin e kushtit të matricës përfundimtare që ka një efekt të dobishëm në veçoritë e konvergjencës të algoritmit parakushtëzues.

Në rastin e parakushtëzimit GMRES pa restart metoda konvergjon ngadalë nëse vlerat më të vogla vetjake janë afër zeros dhe konvergjojnë më shpejt nëse këto vlera nuk janë shumë afër zeros. Në të ardhmen metodat iterative priten të kenë përmirësime për një konvergjencë në sa më pak iteracione.

### Literatura

Ilse C.F. Ipsen & Carl D. Meyer (1997): The Idea Behind Krylov Methods

G. M. Del Corso, O. Menchi & F. Romani (2015): Krylov subspace methods for solving linear systems, 1-5

A.N. Krylov. (1931): On the numerical solution of the equation by which in technical questions frequencies of small oscillations of material systems are determined

W. Auzinger & J.M. Melenk (2017): Iterative Solution of Large Linear Systems; 76-78

Shitao Fan (2018): An Introduction to Krylov Subspace Methods

Y. Saad & M.H. Schultz, (1986): $GMRES$: A generalized minimal residual algorithm for solving nonsymmetric linear systems

Yousef Saad (2000): Iterative Methods for Sparse Linear Systems

C.T.Kelley (1995): Iterative Methods for Linear and Nonlinear Equations, 33-34

Yousef Saad (2011): Numerical methods for large eigenvalue problem